Visualización y simulación de las interacciones de la red de proteínas del potyvirus del tabaco

Alumno: Francisco Salavert Torres (*frasator@inf.upv.es*)
Codirector: Gabriel Bosque Chacón(*gabbosch@etsii.upv.es*)
Director: Doctor Jesús Andrés Picó Marco (*jpico@ai2.upv.es*)

12 de julio de 2015

Índice

1. Introducción

Un sistema de control consta de una serie de elementos que regulan, dirigen, administran y ordenan el funcionamiento de otro sistema, con la finalidad de obtener un resultado de manera óptima.

Por lo general, se usan sistemas de control en procesos de producción industrial, controlando equipos, máquinas o procesos químicos. En el ámbito de la biología de sistemas, las proteínas pueden verse como dispositivos de un sistema de control que regulan mediante interacciones los distintos procesos de la célula.

Las proteínas son macromoléculas vitales tanto a nivel celular como a nivel sistemático y raramente actúan por si solas. Las proteínas se agrupan y organizan para realizar diversos procesos moleculares esenciales.

Numerosos estudios recientes tratan el funcionamiento de la célula como si de un sistema de control se tratara. La comprensión de estos complejos sistemas biológicos se ha convertido en un problema importante que ha dado lugar a una intensa investigación en el análisis de redes, modelado, identificación y predicción de funciones y enfermedades relacionadas con los genes.

Las interacciones entre proteínas o PPIs, se refieren a los contactos físicos establecidos entre dos o más proteínas como resultado de eventos bioquímicos y/o fuerzas electrostáticas. Estas interacciones pueden verse como los elementos de un sistema de control que *encienden* y *apagan* las funciones celulares en los momentos apropiados y coordinan las actividades que permiten el funcionamiento de la célula. Estas interacciones son el motor del interactoma de cualquier célula viva. Sin embargo si estas interacciones son alteradas por la infección de un virus u ocurren de forma errónea debido a una variación genética, provocan distintas enfermedades.

El estudio de redes de interacción es uno de los análisis más comunes en biología de sistemas, ya que permite determinar complejas relaciones biológicas de una manera ordenada y metódica. En los últimos años, estas redes se han convertido en un interesante tema de investigación en el campo de la biología, debido al rápido avance en técnicas de secuenciación de nueva generación (Next Generation Sequencing o NGS).

En las redes de interacción los nodos y las aristas son los componentes básicos. Los nodos representan unidades de la red, mientras que las aristas representan las interacciones entre las unidades.

Algunos ejemplos de redes de interacción biológica son: las redes de interacción entre proteínas (protein-protein), las redes de regulación génica (DNA-protein), las redes de co-expresión génica (transcript-transcript), las redes metabólicas y las redes de señalización celular.

Se dispone en estos momentos de una gran cantidad de información de orígenes y naturaleza muy diversa: desde datos de mutaciones en el ARN viral y su efecto sobre la tasa de éxito reproductivo de una población de virus (una especie de análisis de sensibilidad), pasando por información sobre

la reactividad de muchas de las interacciones entre proteínas del virus y el anfitrión, hasta información cualitativa sobre la relevancia y/o frecuencia de las interacciones.

Estas interacciones se han estudiado desde distintas perspectivas como la bioquímica y la dinámica molecular. Toda esta información proporciona una base para construir redes de interacción de gran tamaño, de las que podemos extraer información para estudiar y entender el funcionamiento de dichas interacciones.

La utilización de estos estudios para el análisis y modelado de sistemas biológicos complejos puede ser clave para proporcionar conocimientos sobre el funcionamiento interno de la célula, su función biológica y el origen de distintas enfermedades.

Dentro del campo de la virología es interesante descubrir cuales son los mecanismos clave en el avance de la infección de un anfitrión. Estudiar la red de interacción de proteínas del virus, la red de interacción de proteínas del anfitrión y como ambas redes interactúan entre sí proporciona una nueva forma de identificar los blancos virales durante la infección.

La familia de virus Potyviridae es una de las más grandes e importantes familias de virus que afectan a las plantas. Actualmente se posee información ómica para gran cantidad de virus de esta familia.

Estudiamos la red dinámica de interacciones entre las proteínas del potyvirus [11] y las de la planta (*Arabidopsis thaliana*). Este tipo de virus sirve de modelo para el estudio de otros virus semejantes, como el de la hepatitis C. Concretamente se ha realizado una implementación del análisis presentados en [1], con la finalidad de poder repetir los mismos análisis para otros organismos y virus.

Usando distintas tecnologías web actuales es posible desarrollar aplicaciones muy versátiles. La web ya no sirve únicamente para mostrar páginas HTML. Hoy en día los estándares web están afianzándose de manera rápida debido a su portabilidad y rendimiento en una gran diversidad de dispositivos. Nuevas funcionalidades son añadidas día a día, pudiendo realizar ejecuciones asíncronas o incluso acceder a la tarjeta gráfica (GPU) para crear elementos tridimensionales.

Las aplicaciones web se están convirtiendo en el estándar para desarrollar aplicaciones que interactúan con el usuario en todas las áreas de conocimiento. La web ofrece una plataforma de código libre, compatibilidad con múltiples dispositivos y sistemas y además no requiere instalación.

La finalidad de esta tesis es el desarrollo de una aplicación web capaz de analizar y visualizar estas redes de interacción, permitiendo estudiar el mecanismo de infección de los virus. La aplicación permitirá visualizar y analizar redes biológicas usando las últimas tecnologías del estándar HTML5 [13].

El trabajo mencionado corresponde a un proyecto a medio-largo plazo, en el que colaborarán el Grupo de Virología evolutiva del *Instituto de Bio-*

logía Molecular y Celular de Plantas (IBMCP), y el Laboratorio de Biología de Sistemas del programa de biología computacional en el *Centro de Investigación Príncipe Felipe* (CIPF). Se extenderá la funcionalidad del software Cell Maps desarrollado en el CIPF.

1.1. Organización de los contenidos de la tesina

En el apartado 1, se introduce el tema de la tesis y se dan algunas definiciones.

En el apartado 2, se especifican los requisitos del software que debe implementarse para abordar el problema descrito en la introducción.

En el apartado 3, se describen las tecnologías existentes en la actualidad que han permitido el desarrollo de esta tesis. Posteriormente, se analizan otros proyectos en el área del análisis de redes.

En el apartado 4, se describe la funcionalidad de la aplicación y como se han implementado los requisitos especificados.

En el apartado 5, se muestra la interfaz gráfica de la aplicación y se describe el manejo de la misma.

En el apartado 6, se repasan las aportaciones realizadas en este trabajo y se proponen diferentes mejoras en los componentes de la aplicación que podrían realizarse en futuros desarrollos.

2. Especificación de los requisitos del software

El objetivo final del trabajo consiste en el desarrollo de una aplicación web que permita visualizar la red de proteínas de un virus junto con la red de proteínas de un anfitrión. Además la aplicación analizará los parámetros de la red de interacción. Para ello deberán realizarse los siguientes objetivos principales:

- Obtener y procesar los datos de interacción de proteínas del virus o *Virus Virus Protein Interaction* (VVPI). A partir de estos datos la aplicación será capaz de generar la red de interacciones entre las proteínas del virus.

- Obtener y procesar los datos de interacción de proteínas de la planta o *Host Host Protein Interaction* (HHPI).

- Partiendo de la red de interacciones de las proteínas de la planta y la red de interacciones de las proteínas del virus, se generará una nueva red de interacción llamada *Virus Host Protein Interaction* (VHPI). Dicha red representa como el virus se introduce en la red de interacciones de la planta y permite realizar un análisis de propagación de forma visual.

- La aplicación permitirá visualizar en un grafo bidimensional la red de interacciones de las proteínas del virus. Además se mostrarán los parámetros de dicha red al usuario.
- Así mismo, el usuario podrá visualizar la *Virus Host Protein Interaction Network* (VHPIN) en forma de grafo tridimensional. En la red se podrán representar las proteínas agrupadas según el número de saltos de interacción, con el fin de mejorar la visualización que aparece en [1], en la que sólo se representan dos niveles.
- Por último se mostrarán los resultados de distintos análisis relativos a las proteínas del virus y sus parámetros en la red. Dichos resultados se presentaran en forma de distintas gráficas.

Para obtener y procesar los datos de la VVPI y HHPI se definen los siguientes objetivos secundarios:

- Definir una estructura de datos para almacenar en memoria, importar y exportar el grafo con la red de interacción VVPI y el grafo con la red de interacción HHPI.
- Leer los ficheros de interacciones usando la API XMLHttpRequest de JavaScript.
- Obtener los parámetros del grafo de la red (número de nodos, número de aristas, número de componentes, media de nodos por componente, diámetro, longitud media del camino mínimo y coeficiente de clustering), para ello se accederá a un servicio web que los calculará.

Para generar la red VHPI se definen los siguientes objetivos secundarios:

- Leer el fichero de interacciones entre el virus y la planta. El fichero contiene las relaciones inmediatas entre las proteínas del virus y la planta.
- Con la información de la VVPIN, la HHPIN y este fichero se genera la VHPI.
- Definir una nueva estructura de datos con la que almacenar, importar y exportar la VHPI.

Para visualizar el grafo bidimensional de la red interacciones del virus se definen los siguientes objetivos secundarios:

- Procesar la estructura del grafo y dibujar los nodos y las aristas en función de los parámetros de la vista seleccionada.
- Implementar algoritmos para posicionar los nodos y las aristas del grafo en pantalla.

- Diseñar una vista con posicionamiento circular de los nodos.
- Diseñar una vista en la que los nodos con más conexiones se sitúan en el centro.

Para visualizar el grafo tridimensional de la red de interacciones del virus y la planta se definen los siguientes objetivos secundarios:

- Procesar la estructura del grafo y dibujar en el espacio tridimensional los nodos.
- Generar una vista mostrando el efecto de propagación de la interacciones de cada una de las proteínas del virus sobre la planta por niveles.
- Posicionar cámaras que proporcionen distintos puntos de vista, situándolas en diferentes ángulos del espacio tridimensional.

Para mostrar los resultados de los distintos análisis se definen los siguientes objetivos secundarios:

- Leer los ficheros de resultados de los análisis y mostrarlos en forma de tabla, permitiendo modificar el coeficiente de relevancia.
- Dibujar las gráficas correspondientes a los análisis topológicos de la VVPI:
    1. Dibujar una gráfica con el grado del nodo asociado a cada proteína del potyvirus en la red.
    2. Dibujar una gráfica con la distribución del promedio de la conectividad de cada proteína con sus proteínas adyacentes.
    3. Dibujar una gráfica con los parámetros topológicos de cada proteína del virus.
    4. Dibujar una gráfica con los parámetros topológicos de cada proteína en relación al grado de su nodo en la red.
    5. Dibujar una gráfica con el grado de distribución de la probabilidad acumulada de cada proteína.
    6. Dibujar una gráfica con la distribución de probabilidad acumulada de los parámetros topológicos de cada proteína.
- Estudiar el efecto de propagación en la VHPI a partir del modelo tridimensional.
- Dibujar las gráficas que comparan el efecto de propagación de las proteínas del virus sobre el interactoma planta.
    1. Dibujar la gráfica correspondiente al índice de Simpson entre pares de proteínas del virus a lo largo de la HHPIN.

2. Dibujar una gráfica con la evolución del índice de Simpson de la proteína HC-Pro.

En la actualidad, ya existe software similar para la visualización y análisis de redes biológicas. En el estado del arte repasamos algunas de estas aplicaciones.

3. Estado del arte

En el estado de arte se proporciona una descripción del Potyvirus. Además se explica que son la interacciones entre proteínas y como se modelizan usando grafos de interacción. A continuación se analizan las aplicaciones más relevantes que permiten la visualización y análisis de redes, algunas orientadas a redes biológicas. Por último se explican algunas de las tecnologías y librerías web más recientes, las cuales han sido utilizadas en el software desarrollado en esta tesis de máster.

3.1. Descripción del Potyvirus

El Potyvirus [11] infecta principalmente a plantas y pertenece a la familia Potyviridae. Un ejemplo de Potyvirus es el *Virus de la patata Y*. El treinta por ciento de virus de planta conocidos pertenecen a este tipo y pueden causar pérdidas significativas en los cultivos agrícolas, pastorales, hortícolas y ornamentales. En esta tesis se estudian las interacciones entre el Potyvirus y la planta *Arabidopsis thaliana*, muy común en estudios biológicos.

3.1.1. Virión

Los viriones de la familia Potyviridae están formados por partículas organizadas en forma de varillas flexibles de filamentos. Su genoma está compuesto por una única hebra positiva de ARN (*single-strand ARN*), el cual está recubierto de un caparazón hecho de una capa de proteína codificada a partir del ARN del propio virus.

Tiene una longitud de 720-850 nm y un diámetro de 12-15 nm con simetría helicoidal. En células infectadas se pueden observar cuerpos característicos de la inclusión.

Figura 1: Virión del potyvirus

3.1.2. Genoma

El genoma del virus consta de un una cadena de ssRNA(+) lineal, de unas 10000 bases cubierta por 2000 proteínas CP, esta cubierta se denomina cápside.

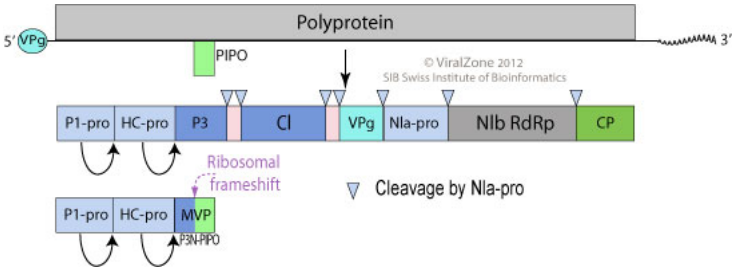


Figura 2: Genoma del potyvirus

3.1.3. Expresión génica

El RNA del virión es infeccioso y sirve como RNA mensajero vírico y genómico. El RNA genómico es traducido en poliproteínas que son procesadas mediante la acción de tres proteasas víricas en elementos funcionales. La proteína P3N-PIPO se expresa mediante un desplazamiento de la P3 y probablemente actúa como una proteína de movimiento.

3.1.4. Proceso de replicación citoplasmático

Cuando se introduce un virión en una célula sana se destruyen una o varias interacciones de la célula del anfitrión lo que lleva a la célula a su destrucción, a continuación el virus usa el material celular remanente para replicarse y generar nuevos viriones, por cada célula se obtendrán unos 100 viriones nuevos que infectarán nuevas células.

El virus penetra en la célula del anfitrión, se libera del caparazón formado por la cápside, liberando el RNA genómico del virus en el citoplasma. El RNA del virus se transcribe para producir una poliproteína que es procesada por las proteasas del virus que la transforman en la proteína RdRp y proteínas estructurales. La replicación ocurre en fábricas de citoplasma viral. Se sintetiza un RNA de doble hebra o dsRNA (*double-stranded RNA*) a partir del ssRNA(+). El RNA de doble hebra se replica proporcionando de esta manera nuevos genomas ssRNA(+) del virus. El ensamblaje del virus ocurre en el citoplasma y la proteína de movimiento P3N-PIPO se encarga de transferir el virus entre las células del anfitrión.

3.1.5. Proteínas del Potyvirus

El Potyvirus esta compuesto de 11 proteínas, a continuación se nombran junto a la función de algunas de ellas:

- HC: Es una proteasa que además esta involucrada en la transmisión por áfidos (pulgones). Interactúa con el factor de iniciación eucariótico 4 (eIF4). Actúa como un supresor de silenciamiento de ARN viral.
- P3: Su función es desconocida. Interactúa con la subunidad de la ribulosa-1,5-bifosfato carboxilasa / oxigenasa.
- CI: Es una helicasa de ARN con actividad de ATPasa. También está involucrada en la unión con la membrana.
- P1: Es una serina proteasa.
- NIa: Se divide en una proteasa y la proteína VPg.
- NIb: Es una ARN polimerasa dependiente del ARN.
- 6K1: No se conocen las funciones.
- 6K2: Se acumula en las membranas celulares del anfitrión y se cree que desempeña un papel en la formación de las vesículas de replicación del virus.
- P3N-PIPO: Se cree que puede estar implicada en el proceso de transferencia del virus entre células del anfitrión.
- VPg: Interactúa con el factor de iniciación eucariota 4E (eIF4E). Esta interacción parece ser esencial para la infectividad viral. Dos proteinasas, P1 y la proteinasa HC catalizan reacciones autoproteolíticas. Las reacciones de división restantes son catalizadas por los mecanismos transproteolíticos o autoproteolíticos y por la proteína de inclusión nuclear (NIA-Pro).

Las proteínas del potyvirus interactúan entre sí y con las proteínas de la célula infectada con la finalidad de replicarse. Estás interacciones se modelan en lo que se conoce como una red de interacción de proteínas.

3.2. Redes de interacción entre proteínas

Las proteínas son macromoléculas vitales para el funcionamiento celular y raramente actúan solas. Diversos procesos moleculares esenciales de una célula se llevan a cabo mediante maquinarias moleculares formadas a partir de una gran número de proteínas organizadas por sus interacciones.

Las interacciones entre proteínas o *Protein-Protein Interactions* (PPIs), se han estudiado desde distintas perspectivas: bioquímica, química cuántica

y dinámica molecular. La información obtenida mediante estas disciplinas ha ayudado a comprender los mecanismos celulares y estudiar el origen de distintas enfermedades.

Las redes de interacciones entre proteínas o *Protein-Protein Interactions Networks* (PPINs) pueden representarse usando teoría de grafos. Un grafo consiste en la representación de unos objetos, en este caso proteínas, por medio de puntos o nodos que se relacionan entre sí mediante líneas o aristas. Las aristas en este caso son las interacciones físicas que se producen entre proteínas. El grafo que representa una red de proteínas de una sola célula, por muy simple que sea la célula, tiene una estructura muy compleja. De ahí la necesidad de tratamientos computacionales para su análisis.

El conjunto de interacciones entre todas las proteínas de una célula recibe el nombre de interactoma. Para poder analizar una red de interacción de proteínas con la teoría de grafos necesitamos una gran cantidad de datos de interacción entre proteínas que las relacionen entre sí. Estos datos de interacción entre proteínas se pueden obtener por medio de distintas técnicas de laboratorio o por medio de recursos informáticos. En los últimos años las tecnologías de alto rendimiento (High-throughput Technologies) han permitido realizar estudios de interacción entre proteínas a gran escala. Entre ellas ha sido especialmente importante la técnica *two-hybrid* que permite detectar *in vitro* interacciones entre proteínas dos a dos a gran escala.

Para que dos proteínas puedan interactuar tienen que coincidir en el tiempo y en el espacio. Hay por tanto que tener muy en cuenta los datos del compartimento celular en el que se encuentran y su momento de expresión para poder elaborar un grafo de interacciones de proteínas (interactoma) que sea correcto. Una vez dibujado el grafo que representa las interacciones entre proteínas se aplican los algoritmos que extraen información sobre las características de la red. En muchos de estos análisis se hace referencia a términos relacionados con la teoría de grafos y que son de muy útil aplicación en el contexto de las redes de interacciones entre proteínas.

Del estudio de redes de proteínas se ha establecido la visión del funcionamiento de la maquinaria celular por módulos. En esta visión modular, las proteínas se asocian de forma coordinada en distintos niveles de organización según requerimientos funcionales. Por ejemplo, la idea de la expresión de un gen para formar una proteína que realiza una función no es falsa pero no es el único modo en que actúa la maquinaria celular. Es muy común la formación de complejos de proteínas que realizan una función. Gracias al estudio de las redes de proteínas, por medio de la teoría de grafos, se han revelado asociaciones entre proteínas que trabajan de forma cooperativa para realizar una función. Esta unidad discreta funcional separable de otras unidades es lo que se conoce como módulos. En la visión modular de la maquinaria celular hay grados de importancia entre los elementos que forman los complejos de proteínas.

Un complejo de proteínas puede cambiar su composición, las proteínas

que siempre forman parte de este complejo formarán el núcleo del complejo. Este núcleo puede interaccionar a su vez con módulos formados por otras proteínas que se unirán a uno o varios núcleos distintos para realizar funciones diferentes, según las condiciones del ciclo celular. En esta visión modular tiene sentido la coexpresión de genes de las proteínas integrantes de un módulo o núcleo. El uso de técnicas computacionales en la elaboración de interactomas es algo novedoso y que aún está en desarrollo, pero es seguro que tener una visión modular de la maquinaria celular puede significar un salto cualitativo en la comprensión de la biología molecular. En el futuro se espera estudiar organismos más complejos.

La gran ventaja de la elaboración del interactoma de una célula u organismo consiste en poder tener una visión de conjunto de la maquinaria celular y por tanto de los procesos fisiológicos, pudiéndose predecir a nivel global las repercusiones que se originan en alteraciones puntuales. Esta capacidad puede darnos muchas ventajas teóricas, por ejemplo: Poder predecir los efectos secundarios producidos por el uso de una droga que actúa aparentemente solo en un elemento de la maquinaria celular muy concreto pero cuyo efecto puede notarse en elementos muy alejados de su objetivo. Poder seguir las alteraciones moleculares producidas por una enfermedad como el cáncer para conseguir estrategias más eficientes en su tratamiento. Incluso elaborar nuevas hipótesis biológicas más completas sobre enfermedades complejas.

3.2.1. Parámetros topológicos de una red

A continuación se describen los parámetros topológicos que caracterizan una red:

Número de nodos La cantidad de nodos que forman parte de una red. Por ejemplo, en la figura 3 hay 6 nodos.

Número de aristas La cantidad de aristas que forman parte de una red. Por ejemplo, en la figura 3 hay 5 aristas.

Número de componentes conexas En las redes no dirigidas, dos nodos están conectados si hay un camino entre ellos. Dentro de una red, todos los nodos que están conectados entre sí forman una componente conexa. El número de componentes conexas indica la conectividad de una red, un número de componentes conexas pequeño sugiere una conectividad mayor. Por ejemplo, en la figura 3 hay dos componentes conexas.

Media de nodos por componente Como su propio nombre indica es la media del número de nodos de todos los componentes de una red. En la figura 3 la media de nodos por componente es $(4 + 2)/2 = 3$.

Diámetro de la red La distancia entre dos vértices de una red se define como el camino de menor número de aristas entre ellos. El diámetro de una red se define como la distancia entre los nodos más alejados de una red. En la figura 3 el diámetro de la red es 3, la distancia entre los nodos 3 y 5 (o los nodos 3 y 1).

Longitud media del camino mínimo La longitud media del camino mínimo, también conocido como la longitud del camino característico, proporciona la distancia esperada entre dos nodos conectados. En la figura 3 la longitud media del camino mínimo es $(1+1+2+1+1+2+1)/7 = 9/7 = 1{,}285$.

Coeficiente de agrupamiento de la red El coeficiente de agrupamiento de una red es la media de los coeficientes de la agrupación para todos los nodos en la red. En el siguiente apartado se explica el coeficiente de agrupamiento de un nodo.

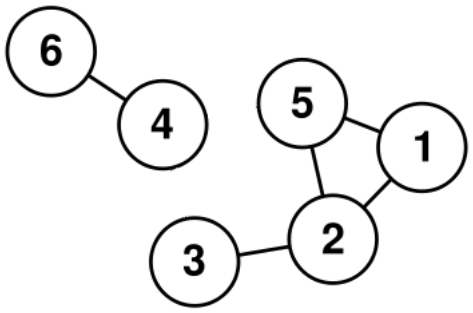


Figura 3: Ejemplo de una red

3.2.2. Parámetros topológicos de los nodos de una red

A partir de los parámetros de la red se pueden obtener parámetros más complejos que nos ayudarán a caracterizar la red de interacción entre proteínas.

Grado de un nodo El grado de un nodo es el número de aristas de la red incidentes al nodo. Por ejemplo, el nodo a de la figura 4 tiene grado 2.

Conectividad de los vecinos La *conectividad* de un nodo es su número de vecinos (nodos conectados por una arista). La *conectividad de vecindario* (*neighbourhood conectivity*) de un nodo n se define como la conectividad promedio de todos sus nodos vecinos. La distribución de la conectividad del vecindario se define como la media de las conectividades del vecindario de todos los nodos n con k vecinos para $k = 0, 1 \ldots$

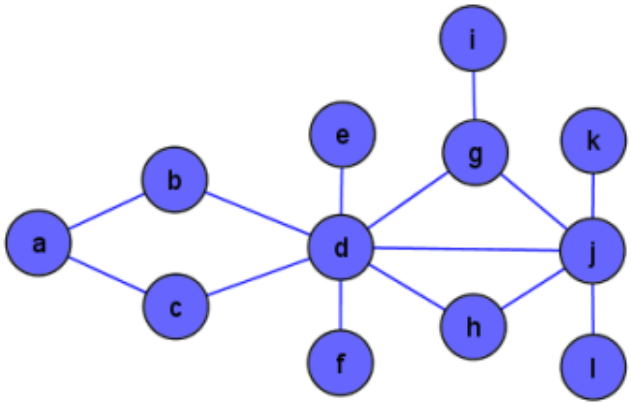


Figura 4: Ejemplo de una red

Esta definición puede extenderse a redes con aristas dirigidas. Al igual que en un grafo dirigido definimos el grado de entrada y de salida de un nodo, en la red definiremos la conectividad de entrada y la conectividad de salida de cada nodo.

En una red de interacción dirigida pueden definirse tres tipos de conectividad de vecindario: tomando la media de la conectividad de entrada, la conectividad de salida o la conectividad tanto de entrada como de salida. Esta última es la que se estudia en este trabajo.

En este apartado tomamos como ejemplo la red de interacción de la figura 4.

En la figura 5 se ha calculado la media de la conectividad de los vecinos agrupando los nodos según el número de vecinos. Por ejemplo, los nodos con un sólo vecino son cinco (i, e, k, f y l). Como solo tienen un vecino su conectividad de los vecinos es la conectividad de su único vecino ($Ci = 3$, $Ce = 7$, $Ck = 5$, $Cf = 7$, $Cl = 5$). Para hallar la distribución se calcula la media de las conectividades de los vecinos, siendo $27/5 = 5.4$ la media de la conectividad de los vecinos de los nodos con un vecino.

Coeficiente de agrupamiento En las redes no dirigidas, el *coeficiente de agrupamiento* (*clustering coefficient*) de un nodo n se define como $C_n = 2e_n/(k_n * (k_n - 1))$, donde k_n es el número de vecinos de n y e_n es el número de pares conectados entre todos los vecinos de n.

En ambos casos, el coeficiente de agrupamiento es una relación de N/M, donde N es el número de aristas entre los vecinos de n, y M es el número máximo de aristas que podrían existir entre los vecinos de n. En un grafo no dirigido cada arista cuenta como dos en esta relación (una en cada sentido). El coeficiente de agrupación de un nodo es siempre un número entre 0 y 1.

La distribución del coeficiente de agrupamiento medio da la media de los coeficientes de agrupamiento para todos los nodos n con k vecinos para

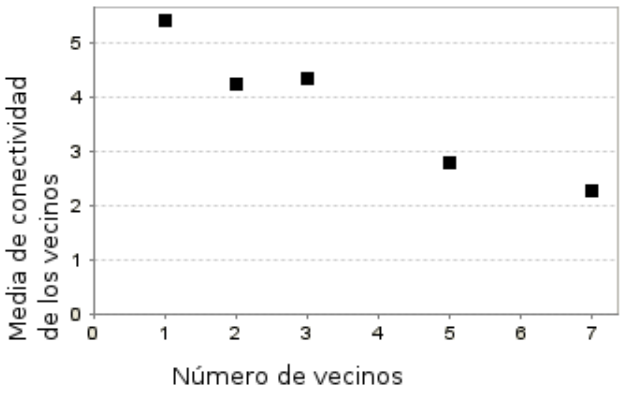


Figura 5: Distribución de la conectividad de la red de la figura 4

$k = 2, 3 \ldots$ Por otro lado, el coeficiente de agrupamiento de la red es el promedio de los coeficientes de agrupamiento de todos los nodos de la red.

El coeficiente de agrupamiento de un nodo es el número de triángulos (3-bucles) que pasan a través de ese nodo, en relación con el número máximo de 3-bucles que podrían pasar a través del nodo.

Por ejemplo, en la figura 6 hay un triángulo que pasa a través del nodo b (el triángulo bcd). El máximo que podría pasar por b sería tres (ya que los pares de nodos (a, c) y (a, d) estarían conectados). Por lo tanto el coeficiente de agrupamiento de b, $C_b = (2 * 1)/(3 * (3 - 1)) = 1/3$.

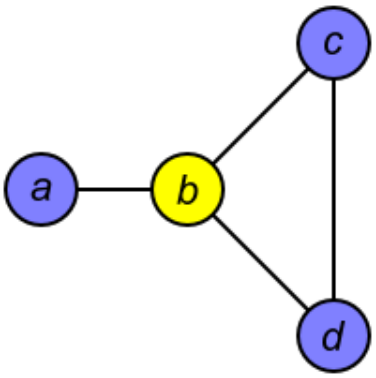


Figura 6: Ejemplo de una red con cuatro nodos y cuatro aristas

Coeficiente topológico El *coeficiente topológico* (*topological coefficient*) de un nodo n con k_n vecinos se define $T_n = avg(J(n, m))/k_n$. Aquí, $J(n, m)$ se define para todos los nodos m que compartan al menos un vecino con n a la hora de calcular la media. El valor de $J(n, m)$ es el número de vecinos compartidos entre los nodos n y m, más uno si hay un vínculo directo entre

n y m.

El coeficiente topológico es una medida relativa del grado en que un nodo comparte vecinos con otros nodos. A los nodos que tienen uno o ningún vecino se les asigna un coeficiente topológico de 0.

Por ejemplo, en la figura 7 se pretende calcular T_b, de esta manera, $J(b,c) = 1+1$, $J(b,d) = 1+1$ y $J(b,e) = 2$. El número de vecinos de b es 3. Por lo tanto $T_b = avg(J(b,c), J(b,d), J(b,e))/k_b = ((2+2+2)/3)/3$.

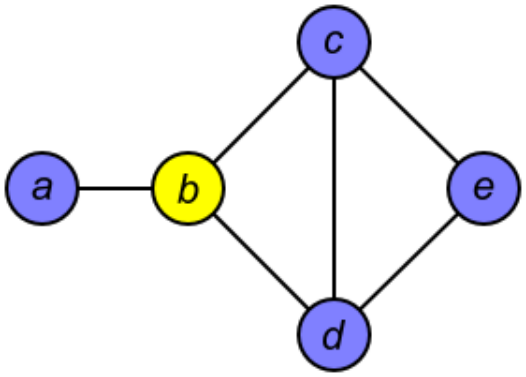


Figura 7: Ejemplo de una red con 5 nodos y 6 aristas

Centralidad de intermediación La *centralidad de intermediación* (*betweenness centrality*) de un nodo n se define como $C_b(n) = \sum_{s \neq n \neq t}(\sigma_{st}(n)/\sigma_{st})$, donde s y t son nodos de la red diferentes de n, σ_{st} denota el número de caminos más cortos de s a t, y $\sigma_{st}(n)$ es el número de rutas más cortas desde s a t que pasan por n.

La centralidad de intermediación se calcula sólo para redes que no contienen múltiples aristas. El valor de intermediación de cada nodo n se normaliza dividiendo por el número de pares de nodos que no contienen n: $(N-1)(N-2)/2$, donde N es el número total de nodos en la componente conexa a la que pertenece n. Así, la centralidad de intermediación de cada nodo es un número entre 0 y 1.

La centralidad de intermediación de un nodo refleja la cantidad de control que este nodo ejerce sobre las interacciones de los otros nodos de la red. Esta medida favorece a los nodos que unen comunidades (subredes densas), en lugar de los nodos que se encuentran dentro de una comunidad.

Por ejemplo en la figura 8 se pretende calcular $C_b(b)$, así pues, $C_b(b) = ((\sigma_{ac}(b)/\sigma_{ac}) + (\sigma_{ad}(b)/\sigma_{ad}) + (\sigma_{ae}(b)/\sigma_{ae}) + (\sigma_{cd}(b)/\sigma_{cd}) + (\sigma_{ce}(b)/\sigma_{ce}) + (\sigma_{de}(b)/\sigma_{de}))/6 = ((1/1)+(1/1)+(2/2)+(1/2)+(0/1)+(0/1))/6 = 3{,}5/6 \approx 0{,}583$.

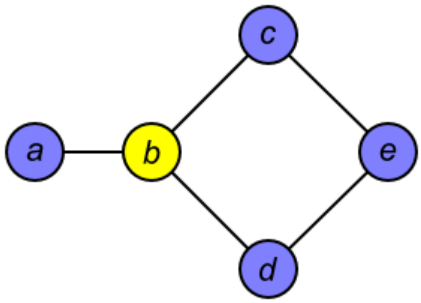


Figura 8: Ejemplo de una red con 5 nodos y 5 aristas

Proximidad central La *proximidad central* (*closeness centrality*) de un nodo n se define como la inversa de la longitud media del camino más corto: $C_c(n) = 1/avg(L(n,m))$, donde $L(n,m)$ es la longitud del camino más corto entre dos nodos n y m. Dado un nodo n, m se define para el resto de nodos alcanzables desde n a la hora de calcular la media. La proximidad central de cada nodo es un número entre 0 y 1.

Un estudio común consiste en comparar la proximidad central de cada nodo con su número de vecinos. La proximidad central de los nodos aislados es igual a 0.

La proximidad central es una medida de la rapidez con la que se propaga la información desde un nodo dado a otros nodos alcanzables de la red.

Por ejemplo en la figura 8 se pretende calcular $C_c(b)$, así pues, $C_c(b) = 1/((L(b,a) + L(b,c) + L(b,d) + L(b,e))/4) = 4/(1+1+1+2) = 4/5 = 0{,}8$

3.2.3. Análisis del efecto de propagación

Entre los distintos tipos de interacciones entre proteínas, existen aquellas producidas por un patógeno (virus o bacterias) en un anfitrión (animales o plantas) y que se representan mediante una red VHPI. Las interacciones entre anfitrión y patógeno se pueden describir a nivel de población (número de individuos infectados por un virus), a nivel de organismo (un virus infecta un anfitrión) o a nivel molecular (un virus se enlaza a un receptor de una célula). Además algunos organismos pueden ser anfitrión y patógeno como las bacterias, que pueden infectar animales y plantas y a su vez ser infectados por virus.

El propósito del análisis entre las proteínas del virus y los anfitriones es entender sus relaciones y como se integran unas con otras, lo cual es fundamental para comprender el proceso de infección. Es importante cuantificar el efecto de cada una de las proteínas virales sobre la red de interacción del anfitrión. A partir de cada proteína viral y siguiendo el interactoma anfitrión se puede calcular la cantidad de pasos consecutivos (interacciones) que se necesitan para llegar a cada proteína del anfitrión. Al final es posible realizar

un mapa por niveles de los pasos entre una proteína viral y la última proteína del anfitrión. Obtener este mapa y representarlo en tres dimensiones ofrece una manera visual de cuantificar el efecto de cada proteína.

3.2.4. Análisis de similitud de propagación

El índice de Simpson (SI) se usa de forma común en biología de sistemas y análisis de redes. Se define como la proporción de nodos conectados relativa al grado del nodo menos conectado:

$$SI(A, B) = |N(A) \bigcap N(B)|/min(N(A), N(B))$$

Donde $N(A)$ son los nodos conectados a A y $N(B)$ son los nodos conectados a B. $N(A) \bigcap N(B)$ son los nodos conectados comunes entre A y B.

El SI cambia en cada interacción y por lo tanto la similitud evoluciona a través de todo el interactoma del anfitrión. Éste índice proporciona una manera rápida de cuantificar la similitud que dos proteínas virales muestran en su relación con el interactoma del anfitrión.

3.3. Tecnologías y librerías

A continuación se explicarán las tecnologías, librerías y entornos usados en la aplicación desarrollada en esta tesis de máster.

Las tecnologías web más actuales permiten la visualización de gráficos tanto bidimensionales como tridimensionales, haciendo posible la modelización y visualización de estas redes.

3.3.1. Tecnologías web

Navegador Web Normalmente un navegador web se entiende como un visor de ficheros HTML, sin embargo hoy en día es capaz de realizar muchas otras tareas, tales como incluir vídeo, audio, gráficos vectoriales y gráficos tridimensionales. Los componentes principales que constituyen un navegador (figura 9) son:

- Interfaz de usuario: Incluye la barra de direcciones, la navegación y el menú de Bookmarks.
- Motor de navegación: enlaza las acciones entre la interfaz de usuario y el motor de dibujado.
- Motor de dibujado: es el responsable de mostrar el contenido, por ejemplo si es HTML, parseará el fichero HTML y lo mostrará en la ventana del navegador.

- Red: Se encarga de ejecutar las peticiones de red HTTP para obtener los distintos recursos de una web.
- Controles visuales: Proporcionan un batería de componentes o widgets, como cajas de texto para formularios, o menús despegables.
- Intérprete de JavaScript [5]: Se usa para parsear y ejecutar el código JavaScript
- Almacenamiento de datos: Esta es una capa de persistencia, se usa para almacenar los datos que pueda necesitar una web, como las Cookies. Actualmente también puede usar mecanismos como localStorage, IndexedDB, WebSQL y FileSystem.

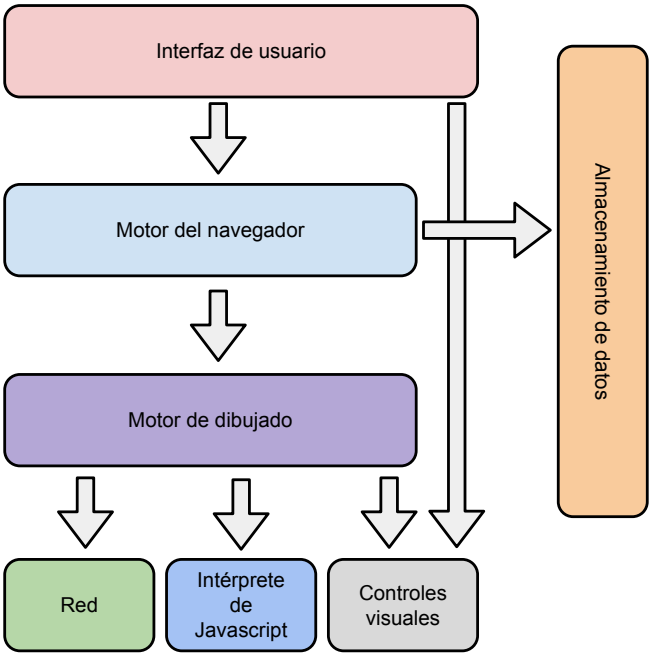


Figura 9: Arquitectura del Navegador

HTML5 Es una de las principales tecnologías de Internet y permite la estructuración y presentación de contenidos en la capa de aplicación. El estándar HTML fue creado en 1990. La quinta revisión del estándar fue establecida por el World Wide Web Consotium (W3C) en Diciembre de 2012.

La nueva API permite desarrollar aplicaciones web más complejas abstrayendo el hardware y el sistema operativo. Por este motivo, HTML5 es una

plataforma con un gran potencial en dispositivos móviles y tabletas. Muchas de sus características han sido diseñadas para su compatibilidad con esta gama de dispositivos de baja potencia.

HTML5 añade nuevas características sintácticas. Algunas de las más importantes son las etiquetas `<video>`, `<audio>` y `<canvas>`, así como la integración de contenido SVG [14] (Scalable Vector Graphics) y el lenguaje de marcado MathML para fórmulas matemáticas.

Un documento HTML se representa en memoria mediante lo que se denomina DOM (Document Object Model). El DOM es un árbol cuyo nodo raíz es la etiqueta `<html>` y cuyas ramas son las subetiquetas del documento.

Una de las APIs más importantes de HTML5 es *XMLHttpRequest* [7], consiste en una clase JavaScript que proporciona una manera fácil de obtener datos de un servicio web mediante URL a través del protocolo HTTP. A pesar de su nombre, *XMLHttpRequest* se usa para obtener cualquier tipo de dato, no solo XML, y soporta otros protocolos además de HTTP como FILE y FTP.

Una vez obtenida la nueva información del servicio web, se puede emplear JavaScript para actualizar partes concretas del documento sin tener que refrescar la totalidad de la página e interrumpir al usuario. Esto se conoce como programación asíncrona o AJAX (Asynchronous JavaScript and XML).

JavaScript Es un lenguaje de scripting usado en desarrollos web. La especificación estándar de JavaScript se conoce como ECMAScript. La sintaxis básica es similar a Java y C++. A partir de 2012 todos los navegadores modernos soportan completamente ECMAScript 5.1.

JavaScript es un lenguaje ligero e interpretado, orientado a objetos y con funciones de primera clase. Se dice que un lenguaje tiene funciones de primera clase cuando estás pueden ser usadas como argumentos. Es un lenguaje de scripting multiparadigma, basado en prototipos, dinámico. Soporta estilos de programación imperativa, funcional y orientación a objetos. JavaScript es empleado como un lenguaje de scripting para páginas web, pero también es usado en otros entornos sin navegador, tales como node.js.

Las capacidades dinámicas de JavaScript incluyen construcción de objetos en tiempo de ejecución, listas variables de parámetros, variables que pueden contener funciones, creación de scripts dinámicos (mediante eval), introspección de objetos (mediante for ... in), y reflexión (los programas de JavaScript pueden generar código en tiempo de ejecución y ejecutarlo).

XMLHttpRequest Es un objeto JavaScript que fue diseñado por Microsoft y adoptado por Mozilla, Apple y Google. Actualmente es un estándar del W3C. Proporciona una forma fácil de obtener información de una URL sin tener que recargar la página completa. Permite al navegador actualizar sólo una parte de la página sin interrumpir lo que el usuario esta haciendo.

XMLHttpRequest es ampliamente usado en la programación AJAX.

A pesar de su nombre, XMLHttpRequest puede ser usado para recibir cualquier tipo de dato, no solo XML, y admite otros formatos además de HTTP (incluyendo file y ftp).

SVG Los Gráficos Vectoriales Redimensionables (*Scalable Vector Graphics*) o SVG son una especificación XML para describir gráficos vectoriales bidimensionales. La gran diferencia con los formatos de mapa de bits es que la información almacenada no son los píxeles de una imagen sino que se almacenan los comandos que describen los trazos de un dibujo, esto permite dimensionar las imágenes sin perder calidad visual.

SVG se convirtió en una recomendación del W3C en septiembre de 2001, por lo que ya ha sido incluido de forma nativa en la mayoría de navegadores web. El estándar SVG incluye tres tipos de objetos gráficos: Elementos geométricos vectoriales (consistentes en rectas, curvas y las áreas limitadas por ellos), imágenes de mapa de bits y texto.

Los objetos gráficos pueden ser agrupados, transformados y recibir estilos comunes. Pueden ser generados antes de renderizarse en la página. El dibujado de los SVG puede ser dinámico e interactivo. El Document Object Model (DOM) para SVG incluye eventos, como *onMouseOver* y *onClick*, lo que permite el desarrollo de gráficos interactivos.

WebGL Forma parte del estándar HTML5 [6]. Esta API sirve para dibujar gráficos tridimensionales en el navegador usando hardware de aceleración gráfica. El acceso directo a la GPU (*Graphics Processing Unit*) desde el navegador abre un gran abanico de posibilidades en el desarrollo de aplicaciones web. El grupo Kronos se encarga de diseñar y mantener WebGL.

Los programas WebGL mantienen el código de control en JavaScript, mientras que el código gráfico es ejecutado en la GPU. El contexto WebGL se instancia dentro de la etiqueta *canvas* de HTML5.

La API de WebGL puede ser bastante tediosa. Existen librerías con el fin de proporcionar más funcionalidad y facilidad de uso. Algunas de estas librerías son Three.js, O3D, OSG.JS y GLGE.

3.3.2. Librerías JavaScript

Three.js Esta librería ofrece una capa de abstracción directamente sobre WebGL, proporcionando una serie de elementos básicos a la hora de implementar un aplicación 3D. Permite generar gráficos en 3D de forma sencilla.

Los elementos principales de three.js son: Renderers, efectos, escenas, cámaras, animación, luces, materiales, shaders, objetos y geometrías.

HighCharts Es una librería JavaScript para dibujar gráficos estadísticos, tales como: diagramas de barras, splines, áreas y gráficos polares (entre otros)

[8]. Esta librería se usa principalmente para mostrar los resultados del análisis de la aplicación.

D3 Es una librería desarrollada en JavaScript para visualizar datos. D3 posee algoritmos para calcular varios tipos de visualizaciones de grafos.

Aunque permite dibujar en el DOM los datos, en el caso de la aplicación no se usará esta funcionalidad. Simplemente se usarán los algoritmos de visualización para obtener las coordenadas de la red. Se han implementado sistemas de dibujado propios: uno en 2D usando SVG y otro en 3D usando Three.js.

JSorolla Es un proyecto que forma parte de la iniciativa OpenCB iniciada por el laboratorio de sistemas genómicos del Centro de Investigación Príncipe Felipe. La finalidad de OpenCB es proporcionar software capaz de manipular, analizar y visualizar datos de secuenciación genómica [12]. OpenCB cuenta con distintos desarrolladores de varias instituciones como el European Bioinformatics Institute.

La librería JSorolla ofrece distintos componentes HTML5 para visualizar datos biológicos. Para el desarrollo de esta tesis he extendido la funcionalidad de esta librería, de la cual soy el principal desarrollador. Existen otras aplicaciones que utilizan esta librería, como por ejemplo Genome Maps y Cell Maps.

3.4. Aplicaciones existentes

Existen distintos tipos de software para visualizar y analizar redes. Algunos están orientados a tratar con cualquier tipo de redes mientras que otros se especializan en redes biológicas.

3.4.1. Gephi

Gephi [4] es un desarrollo del *Gephi Consortium*, una corporación sin ánimo de lucro. *Gephi* es una plataforma de visualización y exploración interactiva para todo tipo de redes y sistemas complejos, grafos dinámicos y jerárquicos. Es una herramienta complementaria a los programas estadísticos tradicionales.

3.4.2. Cytoscape

Cystoscape [3] ha sido desarrollado por el *National institute of General Medical Sciences*, además de otras instituciones colaboradoras como la Universidad de Toronto, la Universidad de California, el Instituto Pasteur, Agilnet Technologies, el Instituto para la biología de Sistemas y el instituto Gladstones.

Cytoscape es un programa de código abierto que permite visualizar redes de interacción molecular y *pathways* biológicos, además integra anotaciones, perfiles de expresión de genes y otros datos. *Cytoscape* fue diseñado inicialmente para investigación biológica, hoy en día es una plataforma general para el análisis y visualización de redes.

Cytoscape viene con una serie de características básicas para la integración de datos, análisis y visualización. Se puede añadir más funcionalidad mediante plug-ins que permiten incluir nuevos análisis, más tipos de ficheros, scripting y conexión con bases de datos usando una API abierta.

3.4.3. Navigator

NAViGaTOR [9] (Network Analysis, Visualization, & Graphing TORonto) es un paquete de software para la visualización y análisis de redes de interacción de proteínas. Puede preguntar a distintas bases de datos para importar redes en 2D y 3D. El software está basado en JAVA, lo cual permite la instalación y ejecución en distintas plataformas.

3.4.4. Cell Maps

Cell Maps ha sido desarrollado en el Laboratorio de Biología de Sistemas del programa de biología computacional del *Centro de Investigación Príncipe Felipe* (CIPF). Es una aplicación web que permite dibujar y analizar redes en el navegador. Esta aplicación se encuentra actualmente en desarrollo. En este trabajo se extenderán las capacidades de Cell Maps.

Basándose en la implementación del modelo de grafos de Cell Maps se ha desarrollado un nuevo módulo de dibujado que permite visualizar redes tridimensionales en el navegador web. Para ello se ha utilizado la reciente capacidad de HTML5 para acceder a la tarjeta gráfica mediante WebGL.

4. Descripción del desarrollo realizado

Una vez especificados los requisitos del software y realizado el estado del arte, en esta sección se explica el desarrollo de la aplicación. Para este trabajo se ha ampliado la arquitectura de Cell Maps, añadiendo nuevas funcionalidades que se describen a continuación.

4.1. Obtener y procesar los datos de interacción de proteínas del virus (VVPI) y de la planta (HHPI)

Los datos empleados corresponden a información de interacción entre proteínas obtenidos de diversas publicaciones.

Es necesario almacenar la información de la interacción: Proteínas involucradas, referencia donde se ha detectado la interacción, especie, localización celular, función y el método de detección utilizado.

Esta información va a crecer con el tiempo, conforme vayan avanzando las investigaciones y aparezcan más resultados acerca de estas interacciones (nuevas especies y métodos de detección de interacciones). Es necesario poder ir ampliando esta información con la finalidad de enriquecer el análisis.

Los datos de interacción entre proteínas se encuentran en ficheros de texto tabulado, los cuales han de ser procesados por la aplicación nada más arrancar. Para cargarlos se ha usado la API de HTML5 *XMLHttpRequest (XHR).*

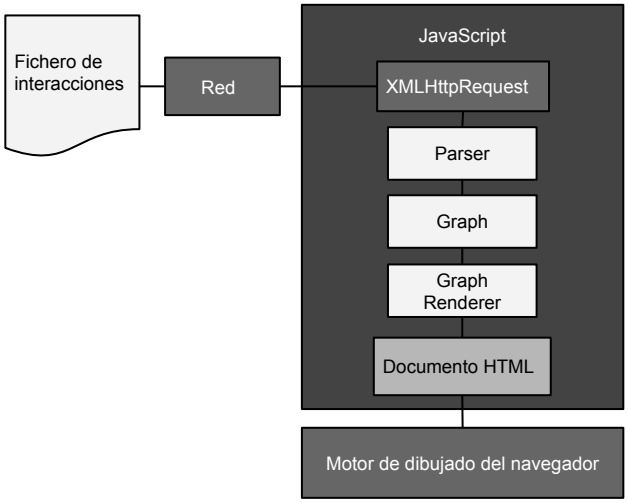
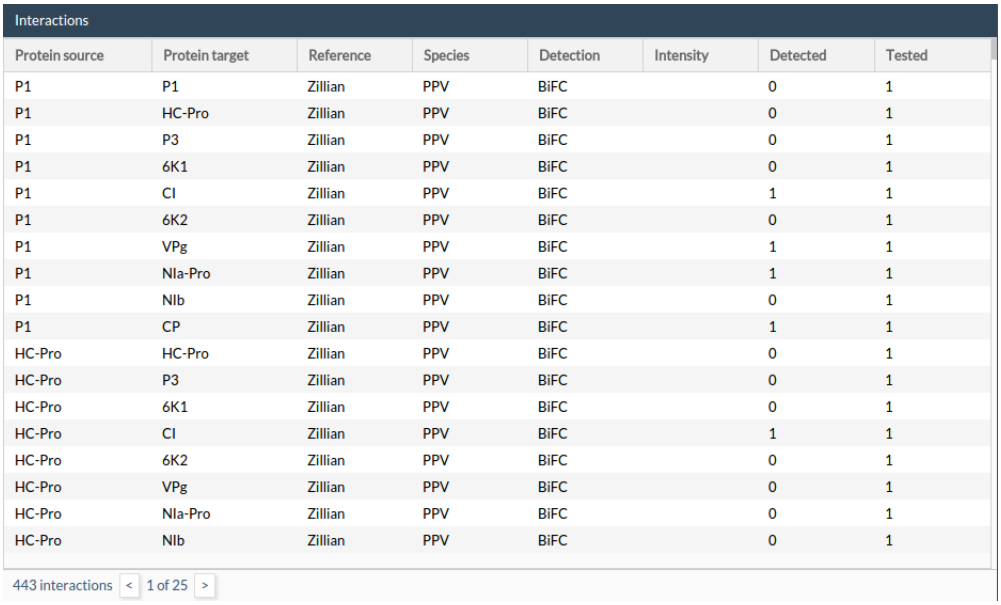


Figura 10: Proceso de importación del fichero y transformación en Graph

4.1.1. Determinar las interacciones relevantes

Una vez cargado el fichero en memoria, se mostraran en una tabla la cual contiene todas las interacciones (figura 11).



Interactions

Protein source	Protein target	Reference	Species	Detection	Intensity	Detected	Tested
P1	P1	Zillian	PPV	BiFC		0	1
P1	HC-Pro	Zillian	PPV	BiFC		0	1
P1	P3	Zillian	PPV	BiFC		0	1
P1	6K1	Zillian	PPV	BiFC		0	1
P1	CI	Zillian	PPV	BiFC		1	1
P1	6K2	Zillian	PPV	BiFC		0	1
P1	VPg	Zillian	PPV	BiFC		1	1
P1	NIa-Pro	Zillian	PPV	BiFC		1	1
P1	NIb	Zillian	PPV	BiFC		0	1
P1	CP	Zillian	PPV	BiFC		1	1
HC-Pro	HC-Pro	Zillian	PPV	BiFC		0	1
HC-Pro	P3	Zillian	PPV	BiFC		0	1
HC-Pro	6K1	Zillian	PPV	BiFC		0	1
HC-Pro	CI	Zillian	PPV	BiFC		1	1
HC-Pro	6K2	Zillian	PPV	BiFC		0	1
HC-Pro	VPg	Zillian	PPV	BiFC		0	1
HC-Pro	NIa-Pro	Zillian	PPV	BiFC		0	1
HC-Pro	NIb	Zillian	PPV	BiFC		0	1

443 interactions < 1 of 25 >

Figura 11: Interacciones del Potyvirus

Dichos datos se pueden filtrar si las interacciones son consideradas irrelevantes, repitiéndose el análisis las veces que haga falta a partir de las interacciones seleccionadas. Se filtrará según el coeficiente de relevancia *RC*, tal y como se describe en el artículo [1].

El RC mínimo se puede seleccionar desde la interfaz de usuario. Aquellas interacciones que pasen el filtro de RC se mostrarán en otra tabla (figura 12). A partir de esta última tabla se construirá el grafo que representará la red de proteínas del virus. Este grafo se modelará usando la estructura de datos *Graph* que se explica en el apartado 4.1.2.

Symetric Interactions RC Threshold: 44

Protein source	Protein target	BiFC detected	BiFC tested	Y2H detected	Y2H tested	Total detected	Total tested	RC %
P1	P1	0	1	1	5	1	6	14
P1	HC-Pro	0	1	1	6	1	7	13
P1	P3	0	1	1	5	1	6	14
P1	6K1	0	1	1	5	1	6	14
P1	CI	1	1	2	5	3	6	57
P1	6K2	0	1	0	5	0	6	0
P1	VPg	1	1	3	6	4	7	63
P1	NIa-Pro	1	1	1	6	2	7	38
P1	NIb	0	1	0	6	0	7	0
P1	CP	1	1	1	5	2	6	43
HC-Pro	HC-Pro	0	1	7	7	7	8	78
HC-Pro	P3	0	1	1	7	1	8	11
HC-Pro	6K1	0	1	0	7	0	8	0
HC-Pro	CI	1	1	2	7	3	8	44
HC-Pro	6K2	0	1	0	7	0	8	0
HC-Pro	VPg	0	1	4	7	4	8	44
HC-Pro	NIa-Pro	0	1	4	7	4	8	44
HC-Pro	NIb	0	1	1	7	1	8	11

58 interactions < 1 of 4 >

Figura 12: Interacciones del Potyvirus filtradas por el Coeficiente de relevancia

4.1.2. Importar y exportar el grafo con la red de interacción VV-PI y el grafo con la red de interacción HHPI

Para modelar los grafos se han definido tres clases *Vertex*, *Edge* y *Graph*, estas clases modelan los nodos, las aristas y el grafo respectivamente.

Estas tres estructuras de datos modelan el grafo y tienen toda la información de los nodos y las aristas (figura 10). Los ficheros tabulados de interacciones se organizan en memoria usando estas clases, tanto en el caso del virus como de la planta.

La jerarquía de estas clases se muestra en la figura 13:

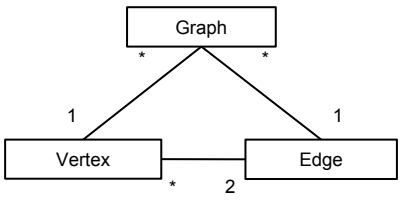


Figura 13: Jerarquía de la clase Graph, Vertex y Edge

En la figura14 se muestran los atributos y métodos de las clases *Graph*, *Vertex* y *Edge* necesarios para trabajar con el grafo.

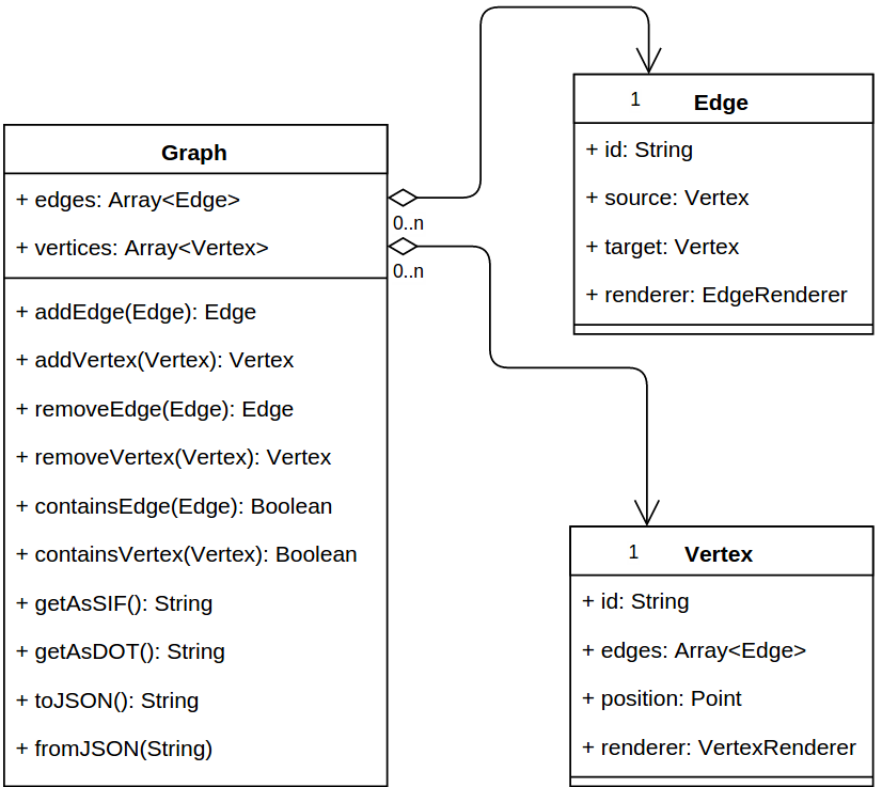


Figura 14: Clases Graph, Edge y Vertex

4.1.3. Lectura y parseo de ficheros

La lectura y parseo de los ficheros se implementa en una clase llamada *NetworkDataAdapter*, cuyo método *parse(fileContent)* recibe una cadena de texto con el contenido del fichero tabulado. Cada línea del fichero contiene la información de una interacción, en la cual se incluye el nombre de la interacción, el nombre del nodo origen y el nombre del nodo destino, tal y como se muestra en el siguiente ejemplo.

```
#Source Relationship Target
P1 vv P1
P1 vv HCPro
P1 vv CP
HCPro vv P1
HCPro vv HCPro
P3 vv HCPro
6K1 vv 6K1
6K1 vv 6K2
```

...

De cada línea se extrae la información de los nodos y las aristas, dicha información se introduce en las clases *Vertex* y *Edge* respectivamente. Al parsear una línea, primero se crean las instancias de los nodos *Vertex* y a continuación se crea la instancia de la arista *Edge*. Si el nombre aparece varias veces se entenderá que el nodo ya ha sido creado y se usará la instancia del nodo previamente creada para construir la arista. Al terminar el método *parse* devolverá una instancia de la clase *Graph* con el grafo de interacción descrito en el fichero.

4.1.4. Obtener los parámetros de la red VVPI

Los parámetros que caracterizan la red en son:

- Número de nodos.
- Número de aristas.
- Número de componentes o número de componentes conexas.
- Media de nodos por componente.
- Diámetro de la red.
- Longitud media del camino mínimo.
- Coeficiente de agrupamiento de la red.

Los parámetros que caracterizan cada proteína son:

- Grado
- Conectividad de los vecinos
- Coeficiente de agrupamiento
- Coeficiente topológico
- Centralidad de intermediación
- Proximidad central

Estos parámetros topológicos tanto de la red como de las proteínas de la red se han implementado en JavaScript según se comentó en detalle en el estado del arte 3.

4.2. Generar de la red de interacción (VHPI)

Para este paso se dispone del interactoma de virus y de la planta previamente cargados en memoria. Además se usará un fichero adicional que contiene el primer nivel de interacciones del virus con la planta. Con estos datos se construye el mapa de interacciones completo.

4.2.1. Leer el fichero de interacciones entre el virus y la planta

Este fichero se cargará de la misma manera que en el paso anterior y se almacenará en una nueva instancia de la clase *Graph* (apartado 4.1).

4.2.2. Generar de la red de interacción (VHPI)

Usando la instancia de *Graph* que contiene la red de interacción de la planta, se añadirán las interacciones del virus y las interacciones inmediatas entre el virus y la planta, quedando así una red conjunta que incluye las interacciones entre las proteínas del patógeno y del anfitrión (figura 15). Para crear la red conjunta se usarán los métodos de *Graph* anteriormente explicados, uniendo las dos redes con las aristas que representan las interacciones inmediatas.

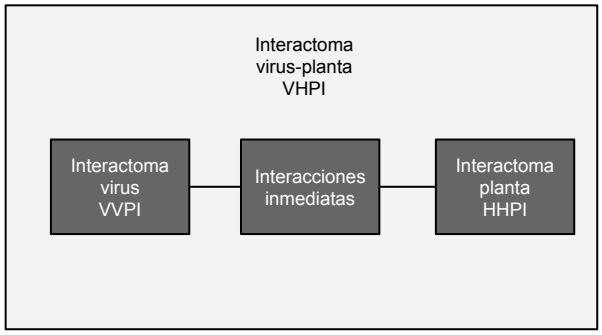


Figura 15: Unión de interactomas

4.3. Visualizar en un grafo bidimensional la red de interacciones de las proteínas del virus VVPI

Para visualizar el grafo del interactoma del virus se usará el módulo de dibujado en dos dimensiones de Cell Maps, este módulo tiene los atributos y métodos necesarios para pintar la red en la ventana del navegador.

4.3.1. Procesar la estructura del grafo y dibujar los nodos y las aristas en función de los parámetros de la vista seleccionada

El módulo de dibujado de Cell Maps llamado *NetworkViewer* está diseñado para ser integrado como librería en otras aplicaciones web. Tomando como entrada un objeto de la clase *Graph* el método *setGraph(graph)* ejecutará las acciones necesarias para mostrar la red.

Los nodos y aristas se dibujan iterando sobre los elementos de la clase *Graph*. Para cada uno de ellos, se crea una etiqueta en lenguaje SVG con la información de dibujado del nodo en sus atributos (posición, color, tamaño...). Las etiquetas de los nodos y aristas son añadidas al documento HTML dentro de la etiqueta principal *<svg>*, la cual indica que el contenido dentro de esta etiqueta ya no es HTML sino una imagen vectorial. Una vez el documento HTML queda modificado, los mecanismos internos del navegador provocan un actualización de la vista del documento en la ventana del navegador.

En la figura (16) se muestran los pasos del proceso de dibujado .

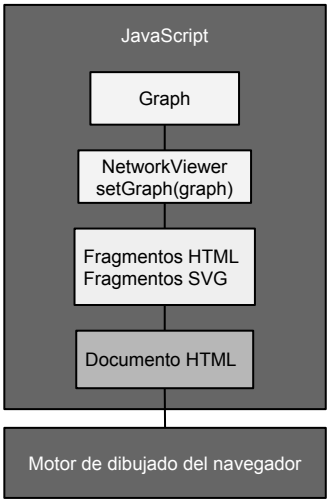


Figura 16: Dibujado de la red en dos dimensiones

4.3.2. Diseñar una vista con posicionamiento circular de los nodos

Se ha implementado un método para posicionar los nodos en forma de círculo, esta visualización resalta los nodos que tienen más conexiones. Para ello hay que posicionar tantos los nodos sobre una circunferencia. Dada la lista de nodos y un radio para la circunferencia, el método devuelve para cada nodo sus coordenadas. Por último se modifican las posiciones en el documento HTML, dando como resultado lo que se observa en la figura 17.

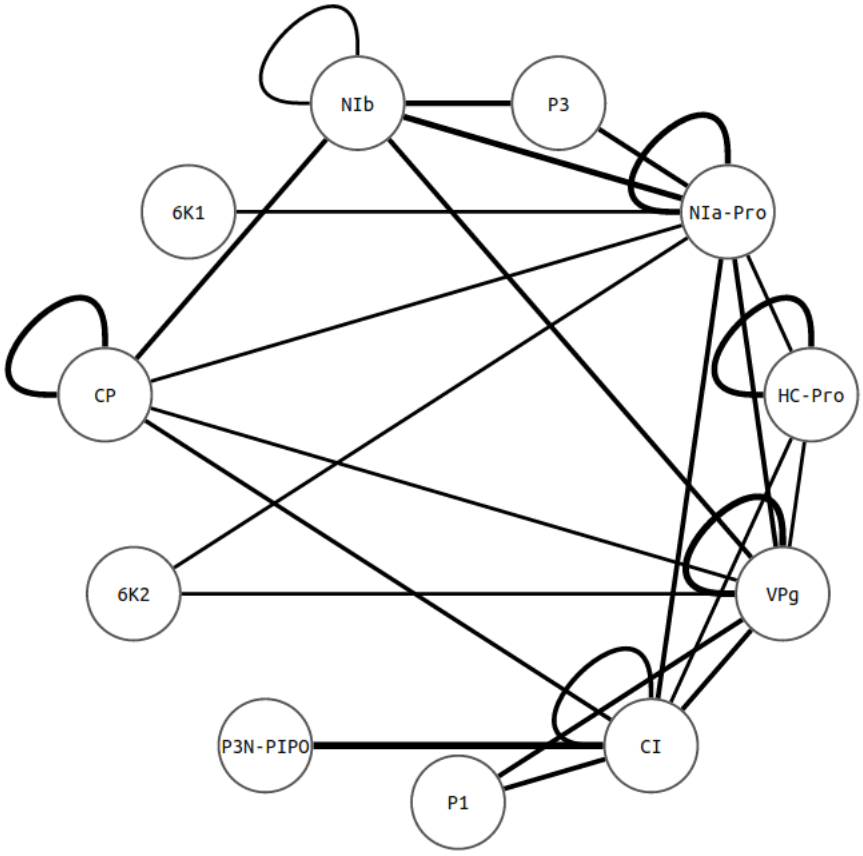


Figura 17: Resultado de la visualización en forma de circulo bidimensional del interactoma del virus

4.3.3. Diseñar una vista Force Layout, en la que los nodos con más conexiones se sitúan en el centro

Existen otras formas de posicionar los nodos en redes biológicas. Cabe mencionar *Force Layout* [2], cuya finalidad es hacer que las aristas tengan más o menos la misma longitud y que se eviten los cruces. Los nodos conectados entre sí se agrupan, dejando ver claramente los grupos de nodos cercanos *clusters*). Esto permite observar fácilmente aquellos nodos con muchas conexiones, ya que aparecerán en el centro de cada grupo.

Para realizar el cálculo de las posiciones se ha usado la librería *D3*, que tiene un método para calcular el *Force Layout* y obtener las posiciones para cada nodo. Este método necesita como entrada una lista de nodos y una lista de aristas. El método devuelve la lista de nodos con las posiciones asignadas. Una vez recuperadas se actualizará el documento HTML. Los resultados se pueden observar en la figura 18.

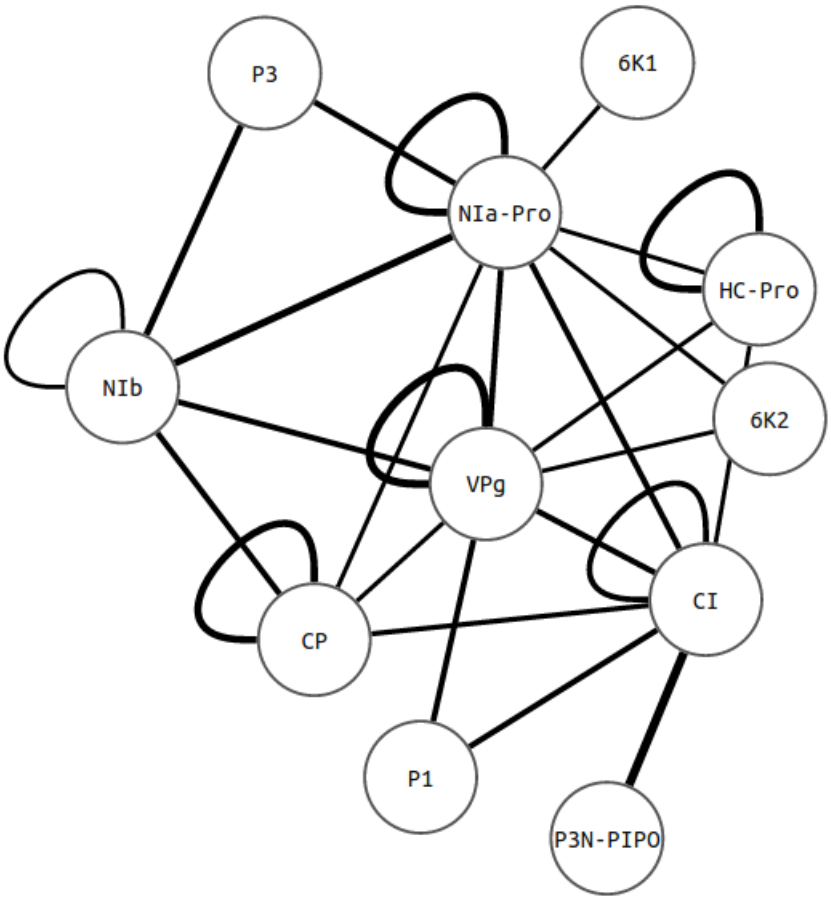


Figura 18: Resultado de la visualización según Force Layout del interactoma del virus

4.4. Visualizar en un grafo tridimensional la red VHPI

Dibujar una red tridimensional proporciona mayor información visual, usar distintos puntos de vista en una escena tridimensional permite apreciar aspectos que en dos dimensiones no aparecen.

Para dibujar la red en tres dimensiones se ha usado la API WebGL que permite acceder a la tarjeta gráfica del ordenador desde un navegador.

A partir del elemento *<canvas>* de HTML se obtiene el contexto necesario para definir una escena tridimensional y agregar objetos como cubos o esferas a la misma.

Se ha utilizado *Threejs* [10] que permite establecer una escena, una cámara, y distintos elementos tridimensionales de manera rápida y sencilla usando JavaScript.

4.4.1. Dibujar en el espacio tridimensional los nodos y aristas de la red VHPI

Para dibujar la red mediante *Threejs* se ha definido una clase o módulo que inicializa el contexto WebGL y crea la escena y la cámara. La escena es una instancia de *THREE.Scene* y almacena los objetos a visualizar.

La clase que dibuja los nodos y las aristas se llama *NetworkViewerWebGL* y tiene definidos los siguientes métodos y atributos para dibujar la red en tres dimensiones.

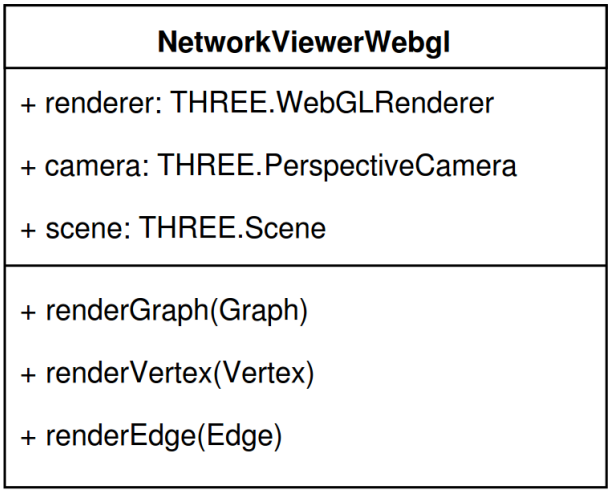


Figura 19: Módulo NetworkViewerWebgl

Una vez definidos estos métodos simplemente con llamar a *renderGraph(graph)* se procesan todos los nodos y aristas del grafo, creando los elementos tridimensionales correspondientes.

Los vértices se han modelado como cubos. Para definir los vértices se ha usado la interfaz *THREE.BoxGeometry(10, 10, 10)* la cual define un cubo con sus tres dimensiones, posteriormente se han establecido las posiciones en el espacio tridimensional modificando la posición de la geometría anteriormente definida.

```
var cube = THREE.BoxGeometry(10, 10, 10)
cube.position.set(vertex.position.x, vertex.position.y, vertex.position.z);
```

Las aristas se han modelado como líneas. Para definir las lineas se ha usado la interfaz *THREE.Line*, la cual necesita tener definidos dos vértices, estos vértices se crean usando la interfaz *Three.vector3*.

```
var edgesGeometry = new THREE.Geometry();
```

```
edgesGeometry.vertices.push(
    new THREE.Vector3(
        edge.source.position.x,
        edge.source.position.y,
        edge.source.position.z
        )
    );

edgesGeometry.vertices.push(
    new THREE.Vector3(
        edge.target.position.x,
        edge.target.position.y,
        edge.target.position.z
        )
    );

var line = new THREE.Line(
    edgesGeometry,
    new THREE.LineBasicMaterial({color: 0xCCCCCC}),
    THREE.LinePieces
    );
```

Por último, estos cubos y líneas se añaden a la escena. Para que aparezcan en la ventana del navegador es necesario llamar a un método de la clase *THREE.WebGLRenderer* llamado *render*, el cual recibe como parámetros la escena y la cámara.

```
    renderer.render(scene, camera);
```

En este momento aparecerán los nodos y las aristas en el navegador (figura 20).

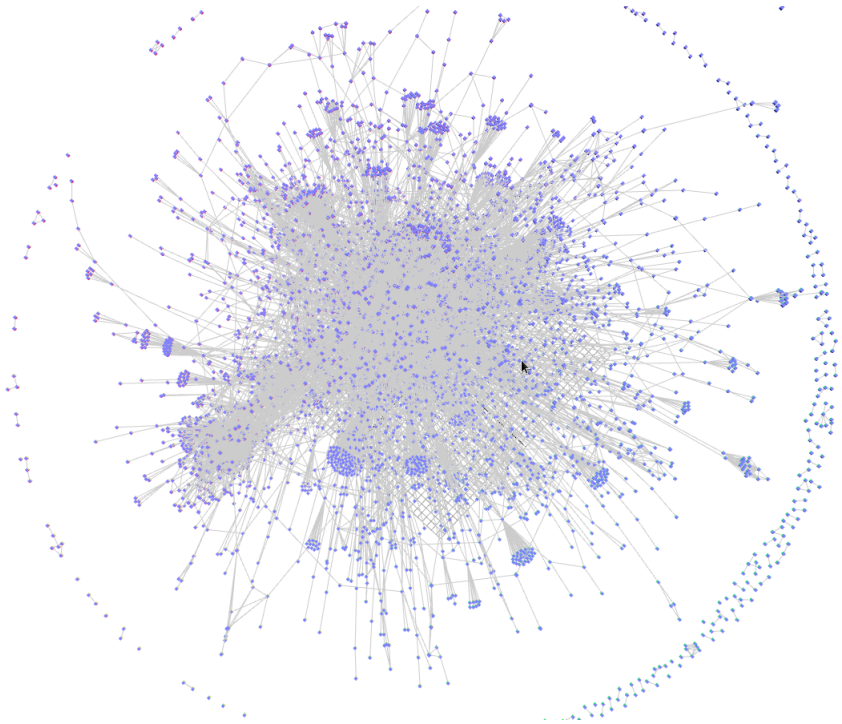


Figura 20: Red tridimensional del VHPI

4.4.2. Crear un sistema de cámaras para mover la vista tridimensional.

Es posible mover la cámara, la cual está situada en la superficie de una esfera imaginaria. La orientación de la cámara es el punto (0,0,0) del sistema de coordenadas de la escena. Al pinchar y arrastrar el ratón la cámara se mueve por dicha superficie, lo que permite ver la red desde distintos ángulos. Usando la rueda del ratón es posible modificar el radio de la esfera lo que implica acercarse o alejarse. En la figura 21 se observan distintos puntos de vista.

4.4.3. Generar una vista por niveles del efecto de propagación

Para visualizar los niveles de interacción se han procesado los nodos de la planta organizando las proteínas por niveles. Partiendo de cada una de las proteínas del virus, el grafo se recorre mediante un algoritmo de búsqueda en profundidad, apuntando en que nivel ha sido visitado cada nodo por primera vez. A continuación se ha asignado una posición Z del espacio a los nodos de cada nivel, posicionándolos en distintos niveles. Los nodos de cada nivel tienen una distribución circular.

Se pueden ver todos los efectos de propagación de las 11 proteínas del

virus sobre el interactoma de la planta *Arabidopsis thaliana* como un mapa interacciones por niveles. En el primer nivel se sitúan las proteínas del virus, se observa como en unos diez saltos de interacción el virus es capaz de alcanzar todas las interacciones de la planta (figura 21).

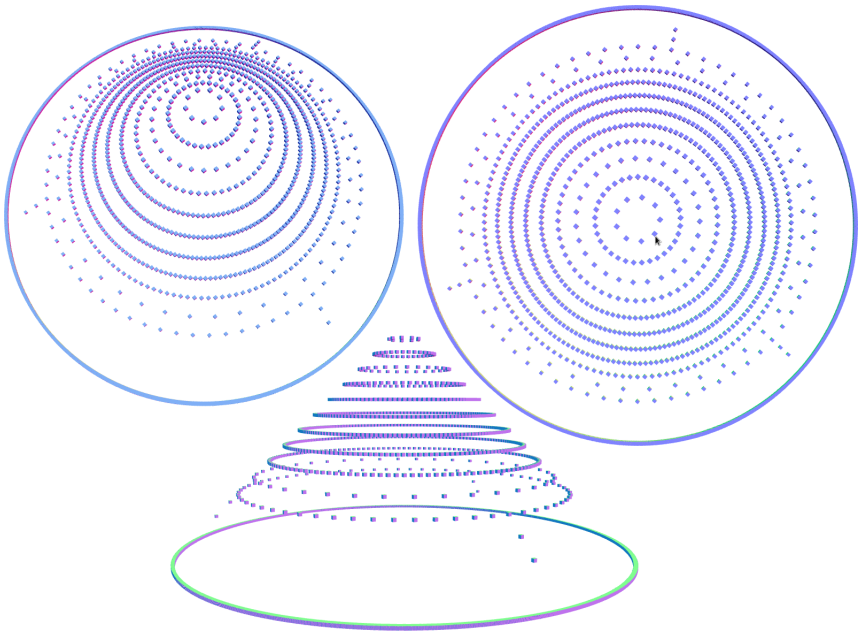


Figura 21: Red tridimensional por niveles del VHPI

Por ejemplo, la proteína P1 establece sólo una interacción con una proteína de la planta (nivel 1), a continuación, esta proteína establece dos interacciones con otras proteínas de la planta (nivel 2, la VHPIN muestra las relaciones de proteína hasta este punto), pero la red sigue creciendo; estas dos proteínas enlazan con 13 proteínas (nivel 3), estas 13 enlazan con 110 (nivel 4) y así sucesivamente.

Las proteínas virales que no interactúan con la planta son: 6K1, CI, 6K2, y P3N-PIPO. Las otras siete proteínas son capaces de alcanzar prácticamente toda la red de la planta (alrededor del 93La propagación comienza en el nivel dos y termina alrededor del nivel ocho. Algunas secciones de la red son inalcanzables porque no están conectadas a la red de interacción principal de la planta. Por supuesto, esto no significa que el efecto viral no sea relevante y significativo en estas partes del interactoma de la planta.

Esta medida de niveles puede ser vista como una variable temporal. El efecto de una proteína viral es probable que sea notado antes en una proteína situada a dos niveles que en una ubicada a seis niveles de distancia.

4.5. Mostrar los resultados de los distintos análisis relativos a las proteínas del virus y sus parámetros en la red.

El siguiente punto consiste en visualizar los resultados de distintos análisis que tienen como fin entender el impacto del patógeno en el anfitrión. Los resultados se dibujan usando *Highcharts*.

4.5.1. Grado de cada proteína del potyvirus

El grado de las proteínas del potyvirus se encuentra en un rango entre 2 y 10, sin embargo se puede distinguir entre las proteínas más y menos conectadas. Las proteínas menos conectadas son P1, P3, 6K1, 6K2 y PN3-PIPO todas en un rango entre 1 y 2. Las más conectadas son HC-Pro, CI, VPg, NIaPro, NIb y CP las cuales se encuentran en un rango entre 5 y 10.

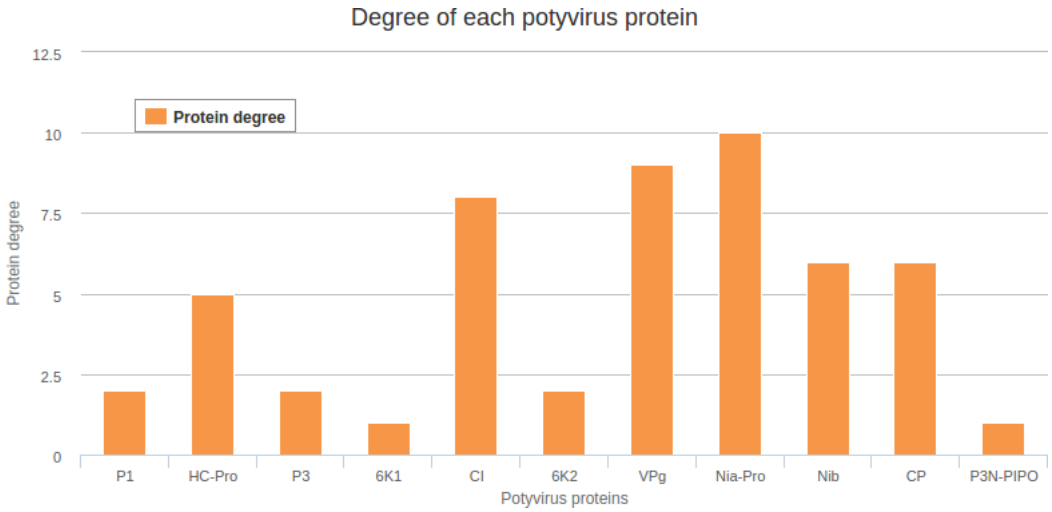


Figura 22: Grado del nodo asociado a cada proteína del potyvirus en la red.

4.5.2. Distribucion de la conectividad de los vecinos

Es interesante estudiar la asortatividad o *assortativity*, que es la preferencia de los nodos de una red por unirse a otros que le son similares en alguna característica. Esto se estudia mirando el grado de cada nodo. En las PPINs se estudia si las proteínas con grado alto tienden a establecer interacciones con otras proteínas de grado alto.

Una forma de estudiar el comportamiento asortativo de una red es observar la conectividad de los vecinos. La conectividad de un nodo es el número de vecinos. La conectividad de los vecinos de un nodo se define como la media de la conectividad de todos sus vecinos. La distribución de la conectividad de los vecinos proporciona la media de las conectividades de los vecinos de todos los nodos con k vecinos, siendo $k = 0, 1 \ldots$ Si la función es creciente la

red es asortativa ya que muestra como los nodos de grado alto, se conectan con nodos de grado alto. Si la función es decreciente, la red no es asortativa ya que los nodos de grado alto tienden a conectarse con nodos de grado bajo.

En la figura 23 los valores de este parámetro decrecen con el número de vecinos, así pues la red de proteínas del virus tiene un comportamiento no asortativo.

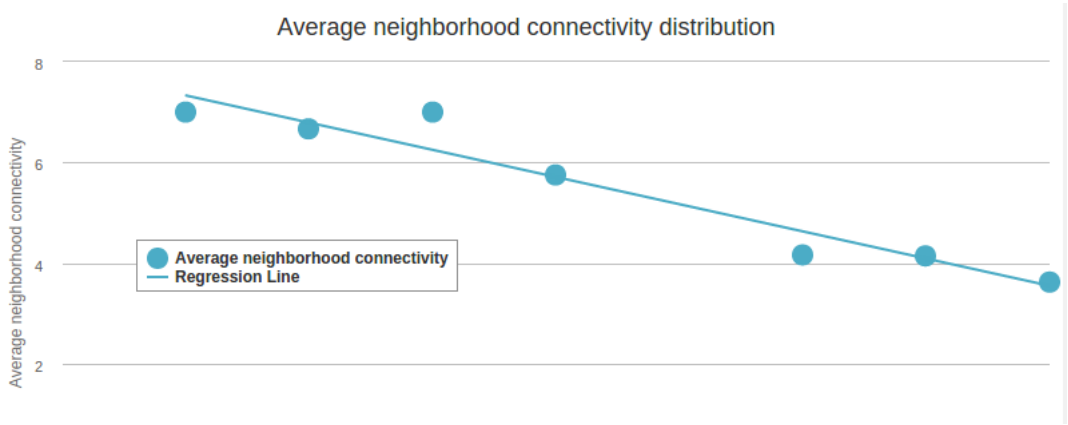


Figura 23: Distribución del promedio de la conectividad de cada proteína con sus proteínas adyacentes.

4.5.3. Parámetros topológicos de cada proteína del virus

Se han calculado cuatro parámetros topológicos: Coeficiente de agrupamiento, coeficiente topológico, centralidad de intermediación y proximidad central (figura 24).

NIaPro, VPg y CI tienen la centralidad más alta y los coeficientes de agrupamiento y topológico más bajos. El coeficiente de agrupamiento y el coeficiente topológico también es bajo en 6K1 y P3N-PIPO ya que no forman ningún triángulo (3-bucles) en la red.

P3N-PIPO esta solo conectado con CI. 6K1 esta solo conectado con NIa-Pro. Los parámetros topológicos de P3N-PIPO y 6K1 son bastante diferentes de las otras proteínas altamente conectadas, presentando un coeficiente de agrupamiento y un coeficiente topológico bastante menor.

Por lo general el coeficiente de agrupamiento y el coeficiente topológico aumentan con el grado mientras que la centralidad y la proximidad decrecen. (figura 25).

Las proteínas menos conectadas tienen un coeficiente de agrupamiento de 0 o 1, mientras que las más conectadas tienen un valor intermedio. La centralidad de intermediación y la proximidad central son mayores conforme aumenta el grado.

HC-Pro se sitúa en un lugar intermedio. Tiene un grado alto y su cen-

tralidad de intermediación es baja. Su coeficiente topológico es alto y su coeficiente de agrupamiento está en el extremo.

El coeficiente de agrupamiento y el coeficiente topológico tienen el peor ajuste en la regresión lineal debido al bajo grado de 6K1 y P3N-PIPO antes mencionado (figura 25).

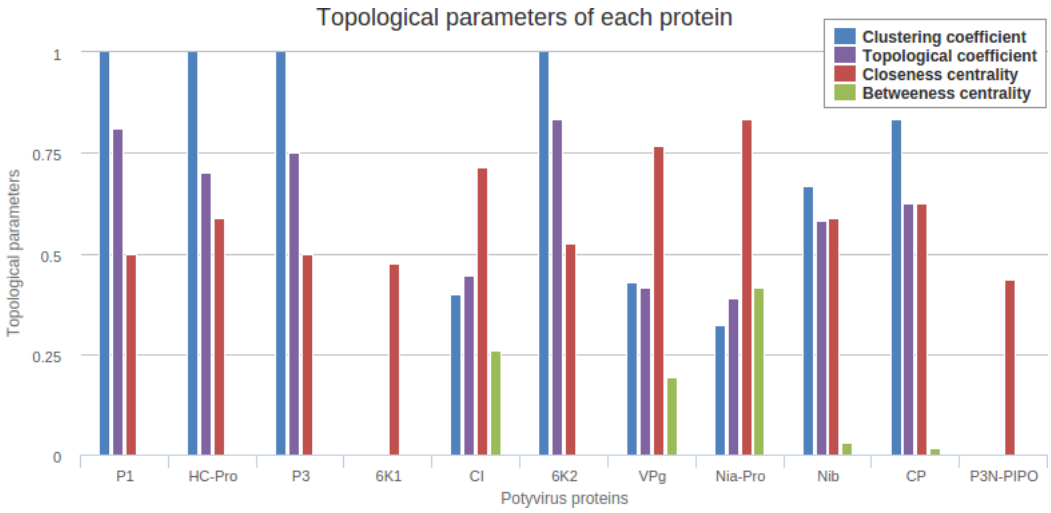


Figura 24: Parámetros topológicos de cada proteína del virus.

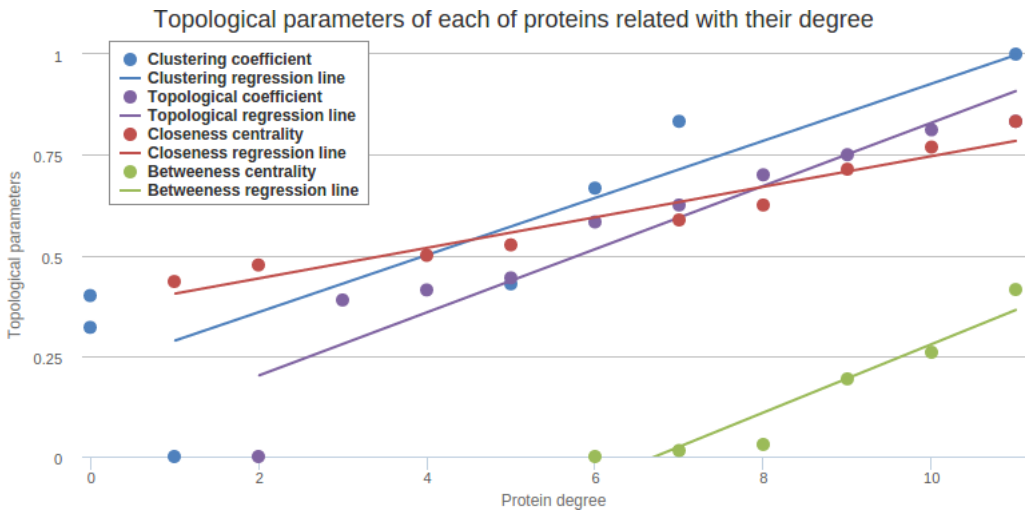


Figura 25: Parámetros topológicos de cada proteína en relación al grado de su nodo.

4.5.4. Distribución de los parámetros topológicos de cada proteína del virus

Se ha calculado la distribución de los diferentes parámetros topológicos. Dichas distribuciones muestran la probabilidad de que un nodo de una red presente un parámetro con un valor particular. Por ejemplo, la probabilidad de que un nodo tenga grado tres.

Cuando se calculan como una distribución acumulada muestran la probabilidad de que un nodo de una red tenga un parámetro menor o igual a un valor. Por ejemplo, la probabilidad de que un nodo tenga grado menor o igual a tres.

Los valores acumulados del grado y de los parámetros topológicos se pueden observar en las figuras 26 y 27. La distribución de la probabilidad acumulada del grado de las proteínas del virus muestra un comportamiento casi lineal, aumentando al aumentar el grado. Las demás distribuciones acumuladas también tienden a ser lineales y crecientes.

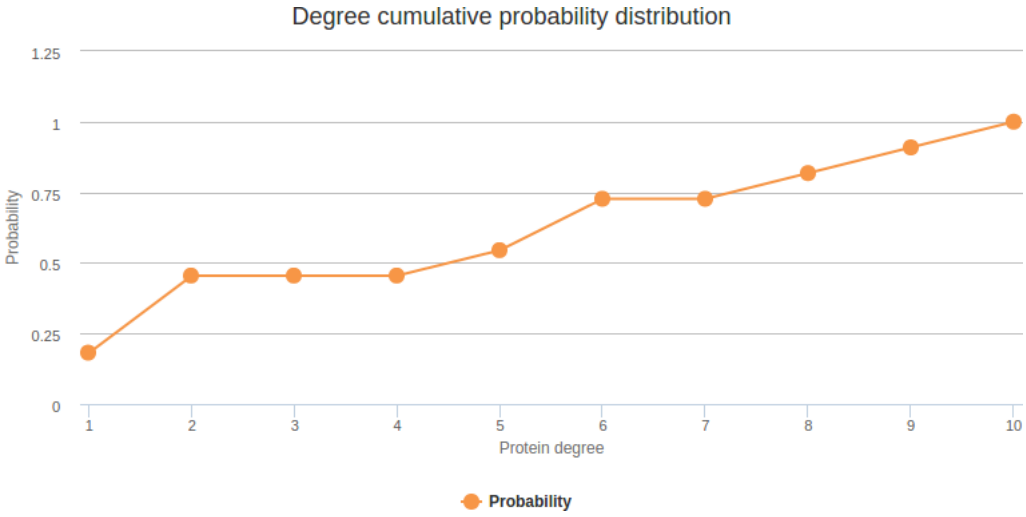


Figura 26: Grado de distribución de la probabilidad acumulada.

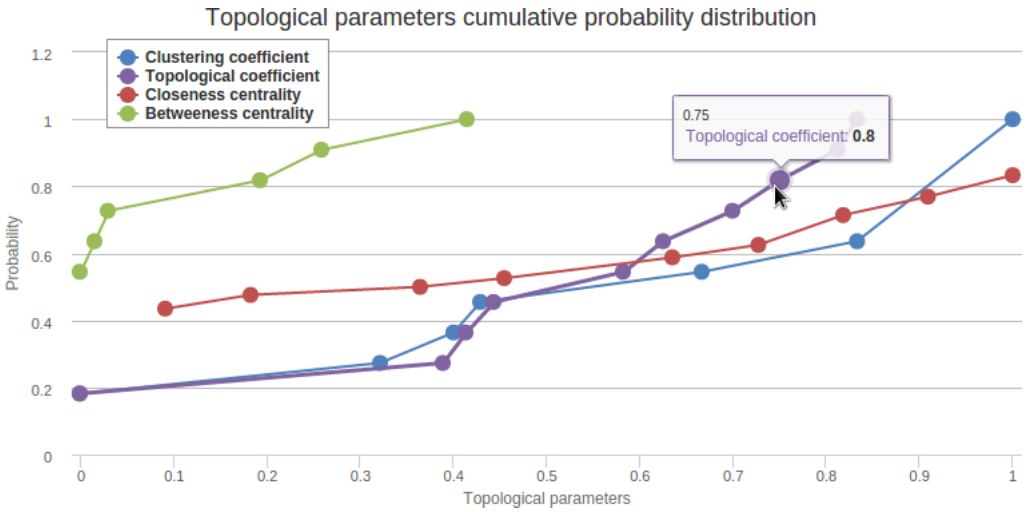


Figura 27: Distribución de probabilidad acumulada de los parámetros topológicos.

4.5.5. Índice de Simpson

Se ha utilizado el índice de Simpson (apartado 3.2.4) para evaluar la similitud entre dos proteínas virales en relación con el anfitrión, observando su manera de interactuar.

Por ejemplo, supongamos que en una nivel determinado P1 llega a cinco proteínas del anfitrión, mientras que HC-Pro llega a 10, y que una de esas proteínas del anfitrión (HP1) es común para ambas proteínas virales. Se forman dos grupos: P1-grupo (con cinco miembros) y HC-Pro-grupo (con 10 miembros). Ambos grupos contienen HP1.

Es posible cuantificar la similitud de estos dos grupos usando un coeficiente de similitud como el índice de Simpson. Éste índice varía entre 0 y 1, expresando la similitud entre dos grupos de proteínas.

El índice de Simpson se calcula para cada par de proteínas y se observa que en 12 niveles se alcanzan todas las proteínas del anfitrión. Al calcularse de manera acumulada en cada nivel (*step*), cada valor en cada nivel da una idea de la similitud. Dibujando todos los niveles en una gráfica se observan comportamientos de propagación parecidos. Tiende a aumentar en los niveles intermedios ya que en ese momento los efectos virales se propagan a toda velocidad, y esas interacciones suelen ser comunes en la mayoría de las proteínas virales.

En la figura 28 se muestra el índice de Simpson para todas las proteínas combinadas con HC-Pro. Esto permite observar comportamientos específicos interesantes. El comportamiento más común entre pares de proteínas es que la similitud empieza en cero y aumenta alrededor del nivel 2-3 hasta que alcanza su máximo en el nivel 7-9.

La primera y la principal diferencia es la velocidad: algunos pares alcanzan un índice de Simpson alto mucho más rápido (HC-Pro, P3) que otros (HC-Pro, P1). Sin embargo, hay algunos casos en los que el índice de Simpson para un par de proteínas disminuye en algunos niveles, como es el caso de (HC-Pro, VPg) que disminuye del nivel 2 al 3. Esto es de alguna manera sorprendente, ya que el índice se calcula con las proteínas acumuladas en cada nivel. Por lo tanto, las redes siempre están aumentando su tamaño en cada nivel. Sin embargo, en algunas interacciones (y para algunos niveles) las redes de ambas proteínas aumentan pero las proteínas del anfitrión comunes a ambas proteínas virales no aumenta proporcionalmente. En consecuencia, existe una disminución absoluta de similitud.

A pesar de todo, el índice de Simpson siempre termina aumentando hasta un valor muy cercano a 1 en este estudio, ya que como sabemos las siete proteínas virales que propagan su efecto llegan a toda la red del anfitrión.

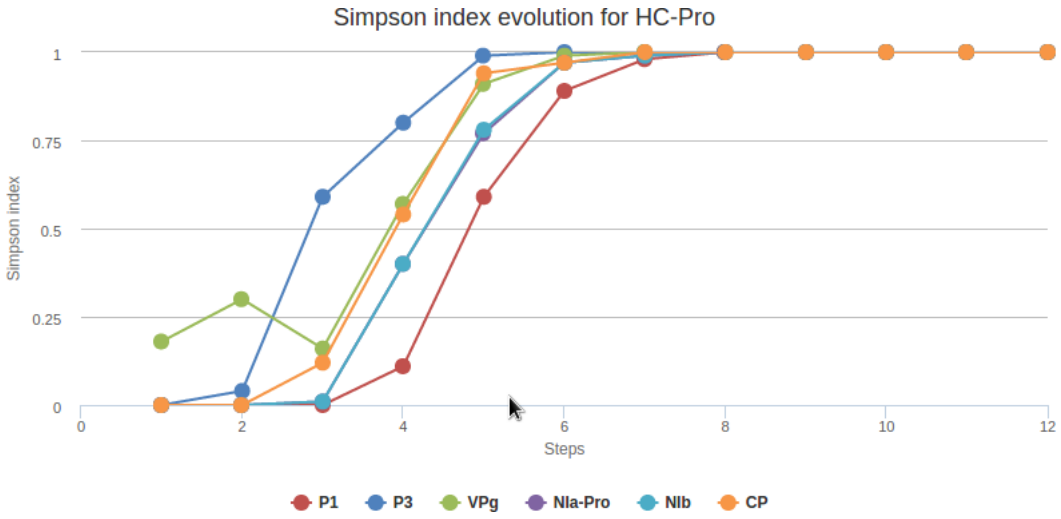


Figura 28: Evolución del índice de Simpson de la proteína HC-Pro.

La información obtenida de este análisis de similitud complementa el estudio de propagación mostrado anteriormente (apartado 3.2.3). Sin embargo, incluso para pares de proteínas representar visualmente la similitud no es trivial. La similitud evoluciona en cada par específico de proteínas según el nivel y mostrar todas las similitudes de todos los pares posibles al mismo tiempo es complejo. Para hacer frente a esto se utiliza una representación basada en una animación de píxeles.

Se ha construido una matriz tridimensional para representar visualmente la evolución del índice de Simpson a través de la red de interacción de proteínas anfitrión-anfitrión (HHPIN). Las dos primeras dimensiones representan los once proteínas virales; esto crea una rejilla que asigna un píxel para cada par de proteínas virales. La diagonal principal no tiene significado biológico

debido a que la similitud de una proteína con sigo misma es siempre uno. Además, la información se repite dos veces en la cuadrícula: (P1, HC-Pro) contiene la misma información que (HC-Pro, P1). El color del píxel representa el valor del índice de Simpson para esa combinación particular. La tercera dimensión es la distancia (medida en niveles) desde el par viral original de proteínas a cualquier punto particular en el HHPIN y se visualiza mediante una animación.

Esta representación (figura 29) permite encontrar rápidamente los puntos de interés: que proteínas virales se unen con el anfitrión, en que niveles cambia la mayoría, que pares de proteínas siguen una evolución determinada, etc.

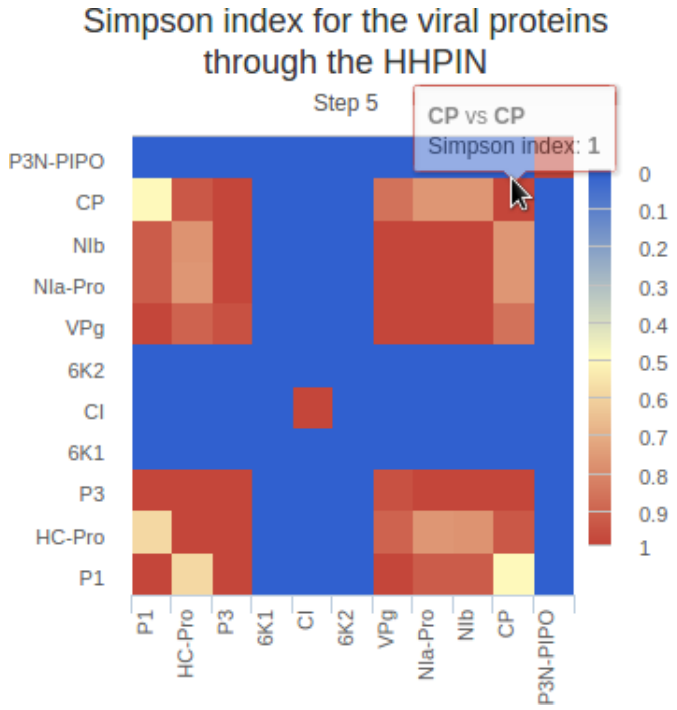


Figura 29: Índice de Simpson de las proteinas virales a lo largo de la HHPIN.

5. Interfaz gráfica de la aplicación

La aplicación tiene una interfaz gráfica compuesta por cuatro secciones, se puede cambiar de una sección a otra usando el menú principal situado en la parte superior.

En la primera sección llamada *Interactions* (figura 30) se muestran las tablas de interacciones conocidas entre las proteínas del virus. En esta sección se permite modificar el coeficiente de relevancia *RC* con el fin del filtrar interacciones.

En la segunda sección llamada *Virus-Virus* (figura 31) se muestra la red

de interacción entre las proteínas del virus, que se obtuvo a partir de la tabla de interacciones filtrada según el coeficiente de relevancia. Las interacciones con un mayor coeficiente de relevancia se representan con un grosor mayor.

En la tercera sección llamada *Virus-Host* (figuras 32 y 33) se muestra la red de interacciones del virus junto con la del anfitrión. Los nodos de color verde indican que son del virus y los de color azul indican que pertenecen al anfitrión. Los nodos se han colocado en distintos niveles, cada nivel indica el número de saltos que hay desde la proteína del virus seleccionada en el panel situado a la izquierda. Estos niveles son el resultado del análisis de propagación. Se pueden añadir y quitar las aristas mediante los botones *Show edges* y *Hide edges*. También se puede elegir entre una visualización circular o de diagrama de fuerzas usando los botones *Force layout* y *Circle layout*. Además, al pasar por encima de cada nodo el ratón aparece un recuadro arriba a la derecha con la información del nodo resaltado. Para los nodos del virus se muestra además dentro de este recuadro un conteo acumulado de los nodos alcanzados en cada nivel.

En la ultima sección llamada *Stats* se muestran los resultados de los distintos análisis relativos a la red proteínas del virus ya mostrados en la sección 4.5. Estas gráficas se generan automáticamente al filtrar la red de interacción del virus con el *RC* (sección 4.1.1).

Figura 30: Interactions

Figura 31: Virus network

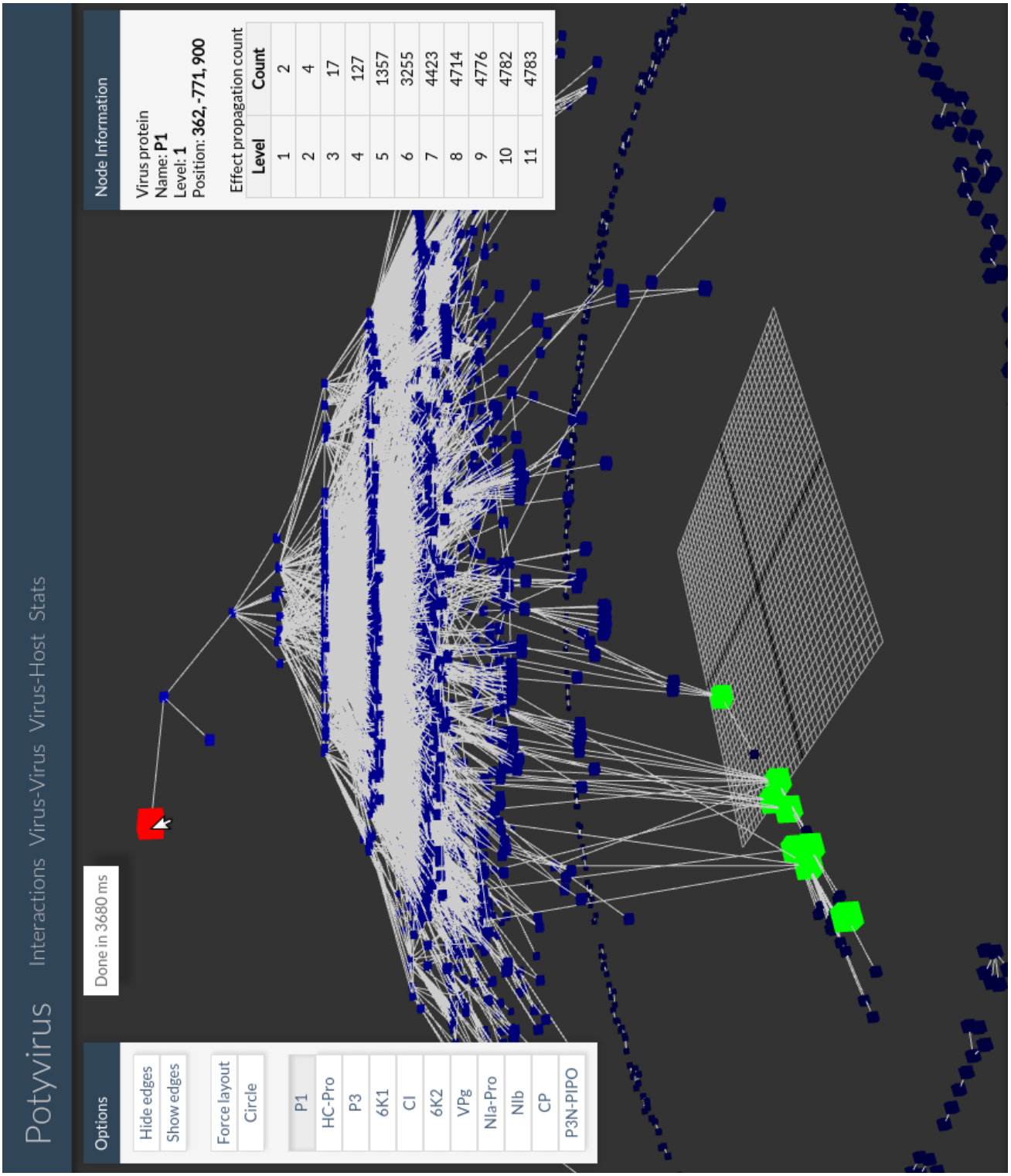


Figura 32: Virus and host network

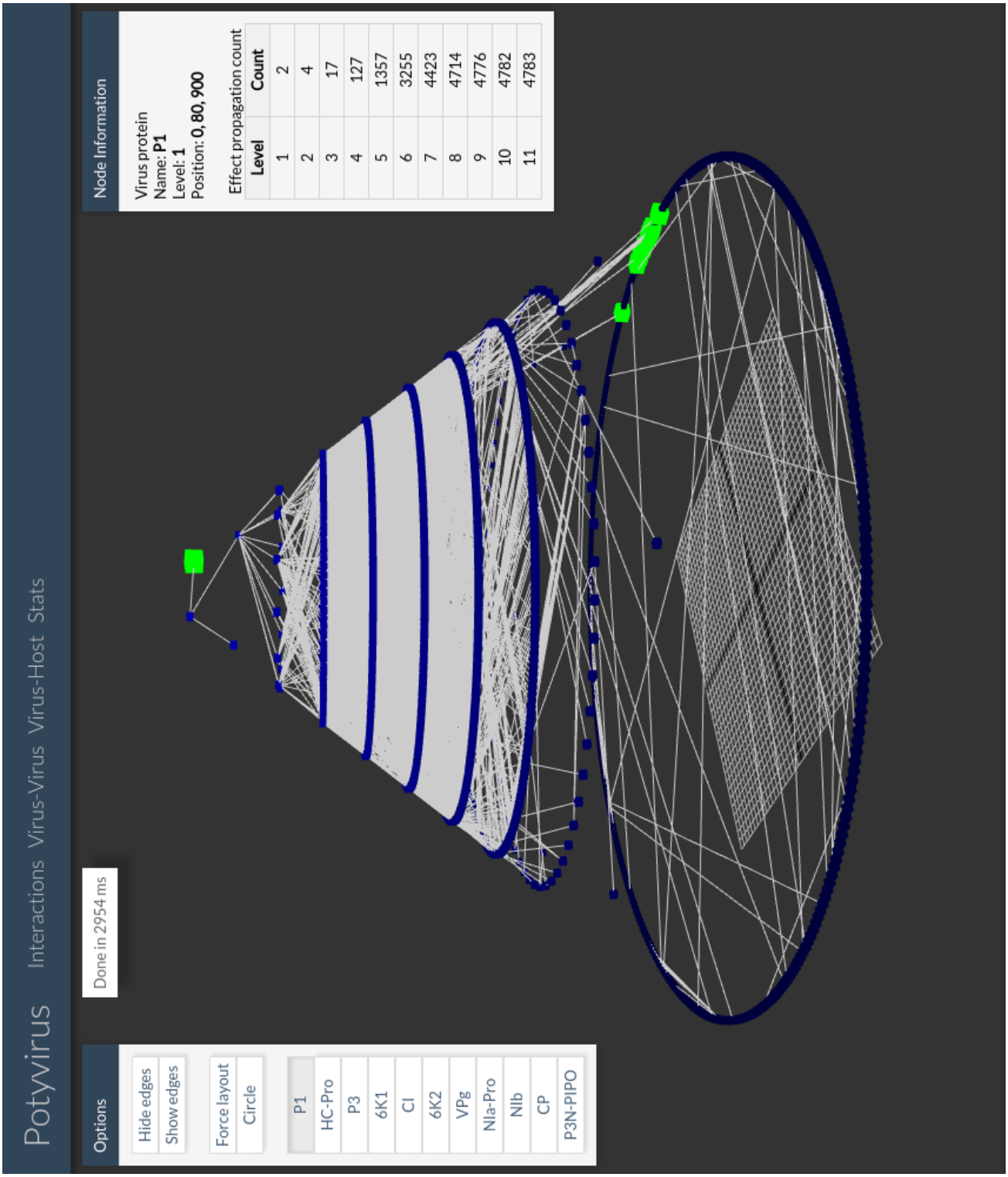


Figura 33: Virus and host network with circle layout

6. Conclusiones y trabajo futuro

La aplicación desarrollada permite ver de manera sencilla como una proteína vírica puede llegar a ser más o menos invasiva. Todo el proceso está automatizado pudiendo repetir el análisis tantas veces como sea necesario con nuevos datos.

El entorno web ha permitido construir una interfaz de usuario para presentar y organizar los resultados usando WebGL y Threejs. Con el lenguaje nativo del navegador *JavaScript* ha sido posible realizar la carga, el procesamiento y el análisis de los datos.

En un futuro se puede mejorar la carga de datos con la finalidad de facilitar la inserción o edición de nuevos datos. También se propone integrar nuevas disposiciones tridimensionales, pudiendo organizar los elementos de la red de maneras distintas.

Otra propuesta interesante sería dibujar en tres dimensiones el análisis de similitud por niveles de dos proteínas del virus.

También se podrían incluir datos de interacción en el tiempo lo cual permitiría estudiar dinámicas.

Por último se propone como trabajo futuro introducir un análisis y visualización de comunidades de proteínas mediante la técnica de análisis de estabilidad de Markov.

Referencias

[1] Gabriel Bosque, Abel Folch-Fortuny, Jesús Picó, Alberto Ferrer, and Santiago F Elena. Topology analysis and visualization of potyvirus protein-protein interaction network. *BMC systems biology*, 8(1):129, 2014.

[2] Mike Bostock. Data-driven documents (d3) force layout. `https://github.com/mbostock/d3/wiki/Force-Layout`.

[3] Cytoscape Consortium. Cytoscape. `http://www.cytoscape.org/`.

[4] Gephi Consortium. Gephi. `http://gephi.github.io/`.

[5] Mozilla developer network. Javascript. `https://developer.mozilla.org/en-US/docs/Web/JavaScript`.

[6] Mozilla developer network. Webgl. `https://developer.mozilla.org/en-US/docs/Web/WebGL`.

[7] Mozilla developer network. Xmlhttprequest. `https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest`.

[8] HIGHSOFT. Highcharts. `http://www.highcharts.com/`.

[9] Ontario Cancer Institute. Jurisica Lab at IBM Life Sciences Discovery Center. Navigator. http://ophid.utoronto.ca/navigator/index.html.

[10] mrdoob. Threejs. https://github.com/mrdoob/three.js/.

[11] SIB Swiss Institute of Bioinformatics. Potyvirus. http://viralzone.expasy.org/all_by_species/50.html.

[12] OpenCB. Josrolla. https://github.com/opencb/jsorolla.

[13] World Wide Web Consortium (W3C). Html5. http://www.w3.org/TR/html5/.

[14] World Wide Web Consortium (W3C). Scalable vector graphics (svg). http://www.w3.org/Graphics/SVG/About.html.